



# **TeamDrive Registration Server Reference Guide**

*Release 5.0.2.0*

**Paul McCullagh, Eckhard Pruehs**



<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 Architecture</b>	<b>5</b>
3.1	TeamDrive Endpoints . . . . .	6
3.1.1	TeamDrive Agent . . . . .	6
3.1.2	TeamDrive App . . . . .	6
3.1.3	TeamDrive Web Portal (Endpoint) . . . . .	6
3.2	Scalable Server Network . . . . .	6
3.2.1	TeamDrive Registration Server . . . . .	7
3.2.2	TeamDrive Host Server . . . . .	7
3.2.3	TeamDrive Web Portal (Service) . . . . .	7
3.2.4	External Authentication Service . . . . .	7
<b>4</b>	<b>TeamDrive Name Server (TDNS)</b>	<b>9</b>
4.1	User data privacy on TDNS . . . . .	9
4.2	General Workflow Between Client, Registration Server and TDNS . . . . .	10
4.3	Blacklisting and Whitelisting Registration Servers . . . . .	10
<b>5</b>	<b>External Authentication</b>	<b>11</b>
5.1	Authentication Service Implementations . . . . .	11
5.1.1	Authentication Service Installation . . . . .	12
5.1.2	Configuration and Setup . . . . .	12
5.1.3	Standard Configuration Parameters . . . . .	12
5.1.4	Authentication Service Customisation . . . . .	13
5.1.5	Example Template Authentication . . . . .	14
5.2	External User Registration . . . . .	14
5.3	External User Data . . . . .	14
5.3.1	Ext Auth ID . . . . .	14
5.3.2	Email Address . . . . .	15
5.4	Compelling Re-login . . . . .	15
5.5	Identifying Externally Authenticated Users . . . . .	15
5.6	Upgrading External Authentication Services . . . . .	16
5.6.1	Upgrading the User Secret Generation Method . . . . .	17
5.7	Implementation Details . . . . .	18
5.7.1	TeamDrive Client: Login Initiated . . . . .	18
5.7.2	Registration Server: “prelogin” Call . . . . .	18
5.7.3	TeamDrive Client: Redirect to External Authentication Service . . . . .	18
5.7.4	External Authentication Service: Login . . . . .	20
5.7.5	Registration Server: “authenticate” Call . . . . .	21
5.7.6	TeamDrive Client: Login Complete . . . . .	22
<b>6</b>	<b>Security and Data Transfer</b>	<b>23</b>
6.1	Encryption in TeamDrive . . . . .	23

6.1.1	AES 256 (Advanced Encryption Standard)	23
6.1.2	RSA 3072 / 4096	23
6.1.3	bcrypt	23
6.1.4	TLS (Transport Layer Security)	24
6.2	Registration	24
6.2.1	The Registration Process	24
6.3	Creating a Space	25
6.3.1	Procedure for Creating a Space	25
6.4	Joining a Space (Accepting an Invitation)	25
6.4.1	Messaging	25
6.4.2	The Space Invitation Document	26
6.4.3	Identifying the Recipient	26
6.4.4	Inviting Unregistered Users	26
6.5	Accessing a Space	27
6.5.1	Object Store Access	27
6.5.2	Host Server Authentication	27
6.5.3	Host Server Reply	27
6.5.4	Space Access Security Features	28
6.5.5	File Publishing	28
6.6	Registration Server Key Repository	28
<b>7</b>	<b>Super PIN Functionality</b>	<b>31</b>
7.1	External Authentication	31
7.2	Account Super PIN Settings	32
7.3	Local Encryption	32
7.4	Requiring Super PIN Activation	32
7.5	Super PIN Repository	32
7.5.1	Recovering from Lost Password	33
<b>8</b>	<b>TeamDrive Client-Server Interaction</b>	<b>35</b>
8.1	Users	35
8.1.1	Create a new user	35
8.1.2	Login as an existing user	36
8.1.3	Forgotten password	37
8.1.4	Check activation	39
8.1.5	Get activation email	39
8.1.6	Undo registration	39
8.1.7	Retrieve user information	39
8.1.8	Retrieve default space depot on a Hosting Service	40
8.2	Devices	40
8.2.1	Invitations	40
8.2.2	Get public key	40
8.2.3	Get device id	40
8.3	Messages, Invitations & Invitation Types	40
8.3.1	Normal invitation	40
8.3.2	Store-forward invitation	41
8.3.3	Invitation for future devices	41
8.3.4	Revoke invitations	41
8.3.5	Delete message	41
8.4	Emails	42
8.4.1	Invitation email	42
8.4.2	Notification email	42
8.5	Change User data	42
8.5.1	Change password	42
8.5.2	Change email	44
8.5.3	License key	44
8.6	Updates	44
8.7	Server URLs	44

8.8	Initial Space Depot Request . . . . .	45
<b>9</b>	<b>Provider Concept</b>	<b>47</b>
9.1	The Provider Code . . . . .	47
9.2	The DISTRIBUTOR File for a Provider . . . . .	47
9.3	User Allocation . . . . .	47
9.3.1	Network Allocation . . . . .	48
9.3.2	Allocation Phases . . . . .	48
9.4	Provider Parameters . . . . .	48
9.5	Hosting Service for each Provider . . . . .	48
9.6	Client License Keys . . . . .	49
9.7	API Access . . . . .	49
<b>10</b>	<b>Account Concept</b>	<b>51</b>
10.1	Members and Managers . . . . .	51
10.2	Using TeamDrive Shop Accounts . . . . .	51
10.3	Adding and Removing Users . . . . .	51
10.4	Account Licenses . . . . .	52
10.5	Account Depots . . . . .	52
10.6	Account Settings . . . . .	53
10.6.1	Department . . . . .	53
10.6.2	Master User . . . . .	53
10.6.3	Advanced Settings . . . . .	53
10.6.4	Depot . . . . .	53
10.6.5	Supported Servers . . . . .	53
10.6.6	Inbox . . . . .	54
10.6.7	Download page for published files . . . . .	54
<b>11</b>	<b>Group Concept</b>	<b>55</b>
11.1	Members and Friends . . . . .	55
11.2	Joining a Group . . . . .	55
11.3	Leaving a Group . . . . .	56
11.4	Type of Groups . . . . .	56
11.5	Group Licenses . . . . .	56
11.6	Group Depots . . . . .	56
11.7	Group Client Settings . . . . .	57
11.8	Group Templates . . . . .	57
11.8.1	Email Templates . . . . .	57
11.8.2	HTML Templates . . . . .	57
11.9	Group Related API Functions . . . . .	57
<b>12</b>	<b>Domains and Services</b>	<b>59</b>
12.1	Domains . . . . .	59
12.1.1	Domain Activation . . . . .	59
12.1.2	Registration using a reserved Email Address . . . . .	59
12.1.3	Domains and Authentication Services . . . . .	60
12.2	Services . . . . .	60
12.2.1	Service Parameters . . . . .	60
12.2.2	Upgrading External Authentication Services . . . . .	61
<b>13</b>	<b>HTML and EMail Templates</b>	<b>63</b>
13.1	HTML Templates . . . . .	63
13.1.1	Activation Pages . . . . .	63
13.1.2	Email Pages . . . . .	63
13.1.3	Portal Pages . . . . .	64
13.1.4	Set Password Pages . . . . .	67
13.2	Email Templates . . . . .	68
13.2.1	Structure of the Mail Templates . . . . .	68
13.2.2	Templates for Client Actions . . . . .	69

13.2.3	Mail Templates for Trial Licenses	71
13.2.4	Mail Templates for User Invite User	72
13.2.5	Mail Templates for Server Administration	72
13.2.6	Mail Templates for API Actions	72
13.2.7	Mail Templates for API License Changes	73
13.2.8	Mail Templates for Account	74
13.2.9	Mail Templates for Groups	74
13.2.10	Mail Templates for Inboxes	74
13.2.11	Mail Templates for Depots	74
<b>14</b>	<b>Settings</b>	<b>77</b>
14.1	Registration Server Settings	77
14.1.1	API Settings	77
14.1.2	Client Settings	78
14.1.3	Email Settings	79
14.1.4	Failed Lookup Control	82
14.1.5	General Settings	83
14.1.6	Proxy Settings	84
14.1.7	Redirect URLs Settings	85
14.1.8	Security Settings	87
14.1.9	TDNS Settings	88
14.2	Provider Settings	89
14.2.1	ADMINCONSOLE Settings	89
14.2.2	API Settings	89
14.2.3	AUTHSERVICE Settings	90
14.2.4	CLIENT Settings	92
14.2.5	CSVIMPORT Settings	95
14.2.6	EMAIL Settings	96
14.2.7	HOSTSERVER Settings	97
14.2.8	INVITATION Settings	99
14.2.9	LICENSE Settings	101
14.2.10	LOGIN Settings	105
14.2.11	REDIRECT Settings	108
14.2.12	SHOP Settings	110
14.2.13	UPDATE Settings	111
14.2.14	WEBPORTAL Settings	111
14.3	Login and Registration Client Settings	112
14.3.1	active-spaces-limit (default: 0)	113
14.3.2	allow-email-login=true/false (default: false)	113
14.3.3	allow-store-forward-invitations=true/false (default: true)	113
14.3.4	allow-webaccess-by-default=true/false (default: true)	113
14.3.5	auto-accept-invitation=true/false (default: false)	113
14.3.6	auto-accept-invitation-mode (default: archived)	113
14.3.7	auto-invite-users=list	114
14.3.8	check-for-updates=true/false (default: true)	114
14.3.9	default-join-mode (default: default)	114
14.3.10	default-publish-expiry-days (default: 0)	114
14.3.11	default-server-mode (default: default)	114
14.3.12	default-server-version-count (default: -1)	114
14.3.13	display-full-name=true/false (default: false)	115
14.3.14	enable-browser-change-email=true/false (default: false)	115
14.3.15	enable-browser-lost-password=true/false (default: true)	115
14.3.16	enable-browser-registration=true/false (default: true)	115
14.3.17	enable-change-email=true/false (default: true)	115
14.3.18	enable-enterprise-server=true/false (default: true)	115
14.3.19	enable-import-server=true/false (default: true)	115
14.3.20	enable-key-repository=true/false (default: true)	116
14.3.21	enable-network-volumes=true/false (default: true)	116

14.3.22	enable-provider-panel=true/false (default: false)	116
14.3.23	enable-publish=true/false/default (default: true)	116
14.3.24	enable-registration=true/false/default (default: true)	116
14.3.25	enable-set-licensekey=true/false (default: true)	116
14.3.26	enable-space-webaccess (default: user-default)	116
14.3.27	enable-tdps=true/false (default: true)	117
14.3.28	enable-webdav=true/false (default: true)	117
14.3.29	fixed-provider-code=true/false (default: false)	117
14.3.30	hash-compare-files=true/false (default: false)	117
14.3.31	inbox-url=URL	117
14.3.32	inbox-user=username	117
14.3.33	master-user=username	118
14.3.34	reg-name-complexity (default: basic-ascii)	118
14.3.35	require-profile=true/false (default: false)	118
14.3.36	scan-enabled=true/false (default: true)	118
14.3.37	spaces-path	118
14.3.38	require-provider-code=true/false (default: false)	119
<b>15</b>	<b>Registration Server API</b>	<b>121</b>
15.1	API Basics	121
15.1.1	API Usage	121
15.1.2	API Input Parameters	123
15.1.3	The <origin> tag	124
15.1.4	The <sendmail> tag	124
15.1.5	Example API Call	124
15.1.6	Error Handling	124
15.2	Registration Server API Calls	126
15.2.1	getsettings	127
15.2.2	loginuser	127
15.2.3	tdnslookup	130
15.2.4	searchuser	131
15.2.5	getuserdata	135
15.2.6	registeruser	140
15.2.7	resendactivation	144
15.2.8	activateuser	145
15.2.9	deactivateuser	146
15.2.10	disableuser	146
15.2.11	enableuser	147
15.2.12	activateclient	148
15.2.13	sendpassword	148
15.2.14	resetpassword	149
15.2.15	changepassword	150
15.2.16	updatepassword	151
15.2.17	setreference	152
15.2.18	setdepartment	153
15.2.19	setemail	154
15.2.20	changeemail	155
15.2.21	confirmnewemail	156
15.2.22	changelanguage	156
15.2.23	updateuser	157
15.2.24	removeuser	158
15.2.25	removedevice	159
15.2.26	deleteuser	160
15.2.27	confirmuserdelete	161
15.2.28	getlicensedata	162
15.2.29	getdefaultlicense	163
15.2.30	createdepot	164
15.2.31	deletedepot	166

15.2.32	updatedepot	167
15.2.33	activatedepot	168
15.2.34	deactivatedepot	169
15.2.35	getdefaultdepotdata	170
15.2.36	gethostfordepot	172
15.2.37	setdepotforuser	172
15.2.38	removedepotfromuser	173
15.2.39	syncdepotdata	174
15.2.40	getdepotdata	176
15.2.41	getspacedata	177
15.2.42	deletespace	179
15.2.43	sendinvitation	180
15.2.44	setinviteduser	181
15.2.45	createlicense	182
15.2.46	createlicensewithoutuser	184
15.2.47	assignusertolicense	184
15.2.48	assignlicensetoclient	185
15.2.49	removeuserfromlicense	186
15.2.50	deactivatelicense	187
15.2.51	activatelicense	188
15.2.52	deletelicense	188
15.2.53	upgradelicense	189
15.2.54	upgradedefaultlicense	190
15.2.55	downgradelicense	191
15.2.56	downgradedefaultlicense	193
15.2.57	getusedlicense	194
15.2.58	setlicensereference	195
15.2.59	removelicense	196
15.2.60	cancellicense	197
15.2.61	setdistributor	198
15.2.62	setcapability	199
15.2.63	wipedevice	200
15.2.64	setlicensecontract	201
15.2.65	setlicenseemail	202
15.2.66	setlicensefeatures	202
15.2.67	setlicenselanguage	203
15.2.68	setlicensetype	204
15.2.69	setlicensevaliduntil	205
15.2.70	resetlicensepassword	205
15.2.71	setlicensepassword	206
15.2.72	changelicensepassword	207
15.2.73	sendtemplatemail	208
15.2.74	createaccount	209
15.2.75	updateaccount	210
15.2.76	deleteaccount	211
15.2.77	addusertoaccount	211
15.2.78	inviteusertoaccount	212
15.2.79	removeuserfromaccount	214
15.2.80	assignaccounttolicense	214
15.2.81	removeaccountfromlicense	215
15.2.82	setdepotaccount	216
15.2.83	removedepotaccount	217
15.2.84	setgroupaccount	218
15.2.85	removegroupaccount	218
15.2.86	getaccountdata	219
15.2.87	creategroup	221
15.2.88	deletigroup	223
15.2.89	inviteusertogroup	224

15.2.90	removeuserfromgroup	225
15.2.91	setgrouplicense	226
15.2.92	removegrouplicense	227
15.2.93	setgroupdepot	228
15.2.94	removegroupdepot	229
15.2.95	userjoinedgroup	230
15.2.96	setgroupclientsettings	231
15.2.97	getgroupdata	232
15.3	Error Codes	235
15.4	User Change Notifications	236
15.4.1	Notification Format	236
15.4.2	Notification Result Handling	238
<b>16</b>	<b>Appendix</b>	<b>239</b>
16.1	Glossary	239
16.2	Abbreviations	239



## INTRODUCTION

The TeamDrive Registration Server stores all required information associated with users of the TeamDrive system. Each user belongs to a “Provider” and may be a member of an account. The Registration Server also records the various TeamDrive client installations (also known as devices), and manages the licences and TeamDrive Hosting Service depots associated with users.

The Registration Server provides a secure messaging service to users for the exchange of the spaces keys that protect the data stored in TeamDrive spaces. For this purpose, the server stores the public key associated with each client device.

The Registration Server is also used to send various notifications via email such as registration, activation and space invitation. The server can also inform users about available client updates.

If you are a Provider or an account manager you can use the Registration Server Admin Console to manager users, settings and other resources belonging to the Provider or account.

Most Registration Servers are part of the TeamDrive Name Server (TDNS) Network which allows users to invite users that are registered on other Registration Servers. All these aspects will be described in detail in the following chapters.



## SOFTWARE COMPONENTS

The TeamDrive Registration Server is based on the following components:

- 64-bit Linux Operating System (Red Hat Enterprise Linux 9 or derivatives)
- MySQL Database Server 8.0
- Apache HTTP Server 2.4
- PHP 8.3 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 autotasks for details.

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



## TEAMDRIVE SYSTEM ARCHITECTURE

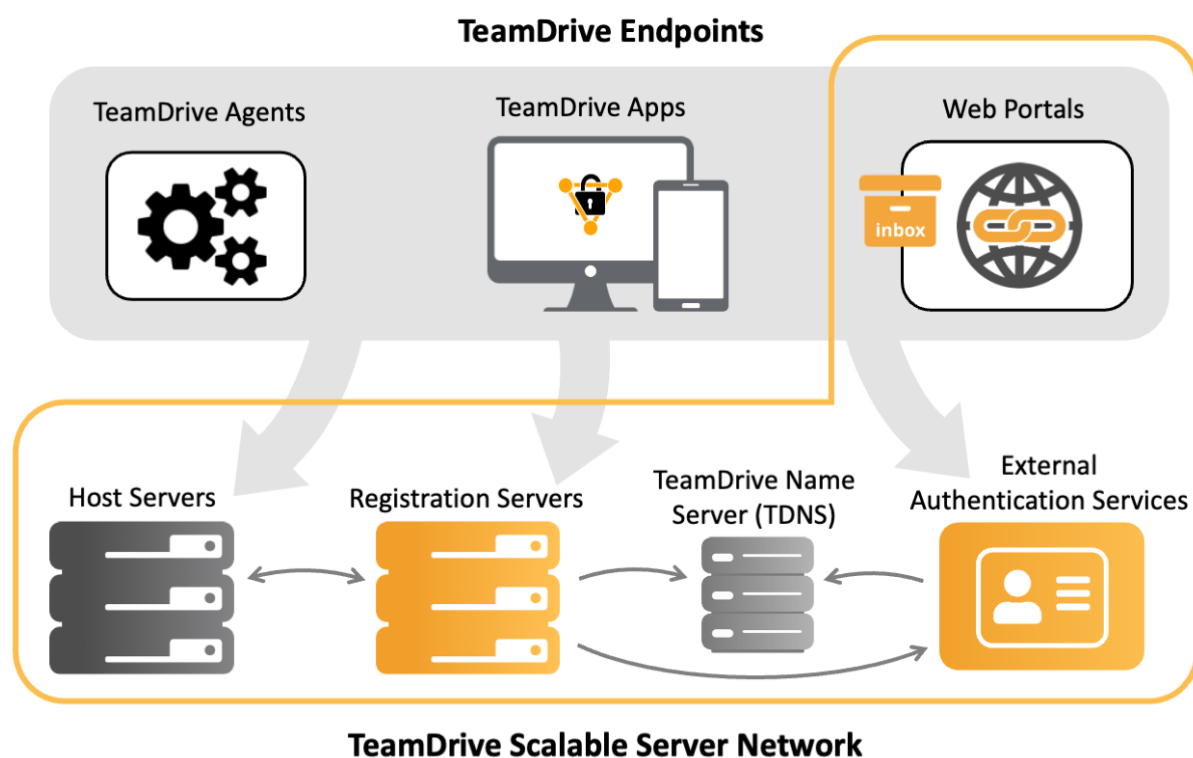
The TeamDrive System consists of the Scalable Server Network and the so-called TeamDrive Endpoints.

TeamDrive Endpoints include instances of the TeamDrive Agent, the TeamDrive App and the Web Portal. Collectively these are also referred to as installations of the TeamDrive Client.

The TeamDrive Scalable Server Network consists of multiple Host Server, Registration Server, External Authentication Services and Web Portals. Whereby the Web Portal functions as both a server and an endpoint.

Also part of the TeamDrive Network is a single instance of the TeamDrive Name Server (TDNS). For more details see *TeamDrive Name Server (TDNS)* (page 9).

All components in the TeamDrive System are illustration below:



In TeamDrive, files are stored in what is referred to as a “Space”. A Space has a number of endusers or members who are also given access to the files. The files are exchanged via a “Depot” on a TeamDrive Host Server.

Each Space has its own 256-bit AES key, which is used to encrypt the files in the Space as soon as the files leave the computer of the enduser. The key is known only by the TeamDrive Client software running on the devices of the members of a Space.

## 3.1 TeamDrive Endpoints

An endpoint in the TeamDrive System is the point where encryption and decryption of data takes place. As an end-to-end encrypted system, all data flowing through TeamDrive is encrypted, and is only decrypted when it reaches an endpoint.

All TeamDrive Endpoints are registered as “Devices” belonging to a particular TeamDrive user on a Registration Server. Each endpoint corresponds to an installation of the TeamDrive Client, either an Agent or an App on a desktop or mobile platform, or as a instance of the TeamDrive Agent running on the Web Portal.

Upon installation of the TeamDrive Client the user is required to register or login to a user account on a TeamDrive Registration Server. Once a device has been registered the user has access to the Host Server Depot (or Depots) that has been assigned to the user. Spaces can be created in the Depot and files uploaded to the TeamDrive Cloud managed by the Host Servers.

invitations to other users to join a Space are sent securely via the Registration Server.

### 3.1.1 TeamDrive Agent

The TeamDrive Agent is service that can be installed on a Mac, Windows or Linux host in order to synchronise data from the local file system to one or more Spaces in the TeamDrive System. In this was the TeamDrive Agent can be used as a component in a automated workflow.

The agent has a Web browser user-interface, which allows for remote access if required.

### 3.1.2 TeamDrive App

The TeamDrive App software is installed on an enduser’s desktop computer or mobile device. Upon installation the enduser must complete a registration or login process. Following installation the user may create Spaces, accept invitations to join a Space and send invitations to Spaces under the users control.

Data is replicated transparently between all devices connected to a Space. On a desktop computer a Space may be mapped to a folder in the local file system, or appear as a virtual directory in the file system.

On mobile devices all spaces are “virtual” which means that the files are loaded transparently on demand, and may be cached locally.

### 3.1.3 TeamDrive Web Portal (Endpoint)

The TeamDrive Web Portal provides access to the TeamDrive system via a browser-based interface. Users can login to a Web Portal and gain access to Spaces of which they are members.

The Web Portal operates by hosting a TeamDrive Agent on behalf of the enduser. A TeamDrive Agent on the Web Portal runs in a secure sandbox provided by the Host System and may employ “local encryption” of the endpoint for additional security (see [Local Encryption](#) (page 32) for details).

## 3.2 Scalable Server Network

TeamDrive is a distributed system consisting of multiple Registration Servers, Host Servers and Web Portals and other components like External Authentication Services.

The TeamDrive Name Server (TDNS) is a central registration point for various services. All server components, other than Host Servers (see below), must be registered on TDNS. Other services (such as a Web Shop) that access the Registration Server API must be also be registered on TDNS.

TDNS also maintains a central directory of usernames and emails of all registered TeamDrive users. All these values are stored as hashes. No user associated data is stored in plain text on TDNS. For further details on TDNS see [TeamDrive Name Server \(TDNS\)](#) (page 9).

The TDNS directory is used to locate the Registration Server associated with a particular user. Once this is established a user can be redirected accordingly, by a Registration Server. Users are also directed to a particular Web Portal and External Authentication Service as required.

As a result, TeamDrive server components can be hosted by independent hosting partners, and also in individual customer datacenters. In this way, customers and partners can maintain complete control of their enduser data. This includes the encrypted data belonging to Spaces as well as backups, and also ensures the privacy of user registration data, and statistical data, such as: Space size and transfer rates.

### 3.2.1 TeamDrive Registration Server

The TeamDrive Registration Server stores information about all registered TeamDrive users. This includes the username, registration email address, a bcrypt password hash and the Public Keys of the enduser. This is the data that is essentially required for the functioning of TeamDrive.

In addition to this, the Registration Server also stores optional profile data provided by the user. This includes the user's language, telephone numbers and a profile picture. This information is encrypted with a key which is only shared with other users via the Space data exchange mechanism. In other words, only users that belong to the same space can exchange profile data.

The email address is verified during initial installation using an activation code as described in the section: *Registration* (page 24).

The Registration Server provides a secure messaging service based on RSA encryption that is used between the TeamDrive Clients. The secure messaging service is used to inviting other users to join a Space (see section: *Joining a Space (Accepting an Invitation)* (page 25)).

One of the Registration Servers is designated as the Master Registration Server. The first Provider created on the Master Registration Server (called the "Default Provider" of the Registration Server) has all privileges to manage the TeamDrive Network. This effectively means the data stored by TDNS is managed on the Admin Console of the Master Registration Server, by users with privileges at the level of the Default Provider.

### 3.2.2 TeamDrive Host Server

The Host Server is responsible for the storage and transfer of changes that occur in Spaces. Each Space is associated with a Depot on a Host Server. The storage and transfer mechanism allows clients to synchronize data, even when the other members of a Space are not concurrently online.

All files stored on a TeamDrive Host Server are encrypted with the 256-bit AES key belonging to the Space.

On installation all Host Servers must be registered on a Registration Server.

### 3.2.3 TeamDrive Web Portal (Service)

As mentioned above the Web Portal is both an endpoint and a service in the TeamDrive System. The users of one or more Providers (see *Account Concept* (page 51)) may be directed towards a particular Web Portal. It is also possible to direct users with an email address of a registered domain name to a particular Web Portal. This way customers can ensure that their users use a designated Web Portal.

The Web Portal service also supports the "inbox" concept as it is implemented in TeamDrive. On the Account level you can create an inbox which can then be used in Spaces to receive uploaded files into specified folders from users that are not necessarily registered TeamDrive users.

### 3.2.4 External Authentication Service

TeamDrive users can be authenticated by an external service such as a corporate LDAP server or Active Directory. In addition, users registered in the Azure or Google cloud, or any service that supports an open protocol such as OAuth2 can all be authenticated by TeamDrive without the need for explicit registration. This is done using a TeamDrive External Authentication Service.

A TeamDrive External Authentication Service is usually hosted by the customer and establishes the link between the TeamDrive components and an authentication service or user registry.

Once the External Authentication Service has been setup, users registered in the external system can login to TeamDrive using their email address and password associated with the external system.

The browser interface of the External Authentication Service can be customized to make it clear to the user where they are logging-in. The user password remains private to the external system, and cannot be intercepted by any of the TeamDrive components.

## **TEAMDRIVE NAME SERVER (TDNS)**

The TeamDrive Name Server (TDNS) is used to manage a distributed network of TeamDrive Registration Servers and Services. For this purpose TDNS stores directories of hashed and encrypted information about the Registration Servers, Providers, Domains, Services/Endpoints and Users in a TeamDrive network.

Primarily, TDNS is used to do the following:

- Ensure that usernames and registration emails are network-wide unique.
- Allow users to be directed to the appropriate Registration Server or External Authentication Service on login.
- Enable invitations to Spaces can be directed to the Registration Server associated with a selected user.
- Manage the creation of Providers on Registration Servers.
- Register and reserve email domains belonging to customers for their exclusive usage in TeamDrive.
- Control which systems (endpoints) are able to access the Registration Server API's.

With one exception, TDNS may only be accessed from a Registration Server. In particular, callers must identify themselves as a Provider (a Registration Server Tenant), and the information they are able to access or modify depends on this identity. The identity is determined using the IP address of the caller and the “Checksum Key” of the Provider. The Checksum Key is generated by TDNS when a Provider is added to a Registration Server. See `tdns-create-provider` for details.

Besides the Registration Server, an External Authentication Service must contact TDNS in order to complete initial setup. In order to confirm the configuration of an External Authentication Service the service must make a lookup using the service name to load and check the identity of service's Registration Server and Provider. Once this has been done, no further calls to TDNS are required.

To enable access to the global TeamDrive instance of TDNS you must allow outgoing access to the HTTPS port 443 for domain `tdns.teamdrive.net`.

### **4.1 User data privacy on TDNS**

User data are not stored in plain text on TDNS. All usernames and email addresses are hashed by the Registration Servers before upload to TDNS.

This ensures that, although Registration Servers are a component of a greater TeamDrive Network, user data (username and email address) remains private and limited to the scope of the Registration Server with which the user is registered.

In addition lookups of this data is only possible from Registration Servers that has identified themselves by IP address and Checksum Key as described above. This means it is not possible to call TDNS directly to determine if a given username or email address is in use by the TeamDrive network, even if the hashing method is known.

## 4.2 General Workflow Between Client, Registration Server and TDNS

The TeamDrive Client/App requires access to the TDNS user directory during login and authentication and when sending an invitation to a Space.

In general the client proceeds by first contacting its preferred or default Registration Server, providing either a username or email address.

The Registration Server will first make a local lookup of the user. If the user is not found, the Registration Server will perform a TDNS lookup. TDNS returns the name of the Registration Server and the Provider Code associated with the user. The Registration Server redirects the TeamDrive Client to new Registration Server.

When redirected, the TeamDrive Client contacts the new Registration Server and sends the original request to this server.

If a user is not already registered on a TeamDrive Registration Server, then the domain name of the email address is used to perform the redirect as required. Domain names can be registered for this purpose as described here: [\*Domains and Services\*](#) (page 59).

## 4.3 Blacklisting and Whitelisting Registration Servers

As an administrator of a Registration Server you are able to determine which of the Registration Servers in the TeamDrive network you trust, and which can be contacted by the your users. This is done using a Blacklist or a Whitelist (see `manage_servers` for more details). This information is stored on TDNS.

Users are restricted from sending and receiving Space invitations to and from users on a distrusted Registration Server.

By default all Registration Servers are trusted, and a Blacklist is used to specify the Registration Servers by name that are not trusted.

This can be changed to Whitelisting whereby the default is then that all Registration Servers are distrusted, and you have to explicitly place all trusted Registration Servers, by name, on a Whitelist.

## EXTERNAL AUTHENTICATION

TeamDrive supports external authentication. If used, the authentication data is not located on the Registration Server. External authentication is performed by a Web-site called an “External Authentication Service” TeamDrive provides a number of standard implementations for Authentication Services, including: Azure, Google, LDAP, Microsoft Activate Directory, Vasco IDENTIKEY Authentication and standard Oauth 2.0 authentication. These will be referred to collectively as “Authentication Providers”.

Services such as LDAP and AD access company resources and therefore must be deployed within the corporate network. This associated Registration Server, on the other hand, may be located anywhere on the internet. Security is ensured by exposing only 2 access points: the login and verify URLs (see the description of **Login URL** and **Verify URL** below).

The HTML pages exposed by the External Authentication Service can be customised with the to reflect the CI (Corporate Identity) of the owner of the service.

All external authentication services must be registered on the Registration Server, see: [Services](#) (page 60).

To support external authentication, the mobile TeamDrive App (and older desktop Clients) provide access the External Authentication Service using an embedded browser. In this case the login is displayed in an alternative panel of the login dialog, following the entry of the user’s email address.

The latest TeamDrive Desktop Clients launch a local browser to access the External Authentication Service. This method uses “session-based” authentication. The Desktop Agent UI or the Web Portal redirect to the External Authentication Service as necessary.

In the first step of the login process the user must enter their email address. If the user is not already know to the Registration Server, the domain of the email address is used to determine the associated External Authentication Service. Login with an email address, rather than a “username”, is therefore required for login using an External Authentication Service.

Contact TeamDrive support in order to receive assistance in setting up your own an External Authentication Service.

If you are running external authentication services setup prior to version 4.5.1 (12 April 2020) of the Registration Server, then “unnamed” services must be upgraded to a named authentication service. See [Upgrading External Authentication Services](#) (page 16) below for details.

### 5.1 Authentication Service Implementations

If you need to use an Authentication Service, TeamDrive provides a number of implementations as mentioned above.

If an implementation is not available for your authentication service, then it is possible to create a custom implementation using the “example implementation” as a template (see below).

### 5.1.1 Authentication Service Installation

An authentication service can be deployed on most Linux systems that supports Apache and PHP 8.3 or later. All TeamDrive implementations have been tested on CentOS 9 and therefore this is the recommended system.

The host providing the authentication service needs to be reachable by the TeamDrive Clients via HTTPS (TCP Port 443) as well as by your Registration Server via HTTP (TCP Port 80), if both systems are in a trusted environment, otherwise HTTPS (TCP Port 443).

Install the External Authentication Service package as described in `install-ext-auth`. This package is also installed on the TeamDrive Registration Server Virtual Appliance (in directory `/var/www/html/authservice`), however, for security reasons, an authentication service should run on the same host as the Registration Server.

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

### 5.1.2 Configuration and Setup

After installation duplicate the `*_config.php.example` file of the service you wish to use, and rename it to `*_config.php`.

You must set the following configuration parameters: `$service_name`, `$reg_server_name`, `$provider_code` and `$allowed_origins` which are common to all authentication service (see below for details of these settings).

Below these settings you will find a section of settings that are specific to the type of service, LDAP, OAUTH2, etc. Set these parameters as required by your service. Detailed comments in the config file describe what is required for each parameter. For AD/LDAP setup, see `:ref:ldap_parameters` for further details.

When the service name has been determined, and other settings such as Registration Server and Provider, you must register the External Authentication Service on the Registration Server. This must be done before you can test the service because, when called for the first time, the authentication service will attempt to verify the existence of the service on TDNS (the TeamDrive Name Server).

To register an authentication service you must enter 2 URLs: the **Login URL** and the **Verify URL**. The **Login URL** must reference the `*_login.php` page of your service, and the **Verify URL** must reference the `*_verify.php` page of your service. See [Services](#) (page 60) for more details on how to registered an authentication service.

The **Login URL** is the start page for user's using the authentication service. For testing purposes you can call this page directly from the browser. During TeamDrive login the TeamDrive Client will redirect to this page to begin the authentication process. Depending on the External Authentication Service, this page will then either display a login dialog, or redirect to the login page of the Authentication Provider (Google, Azure, etc.).

On successful login, **Login URL** page issues an "Authentication Token" which can be verified by the Registration Server using the **Verify URL**. For testing purposes, set `$enable_debug = true` in the configuration file, and you will then be able to check the verification page, ater login. But, remember to remove this debug before you system goes live.

### 5.1.3 Standard Configuration Parameters

There are a number of configuration parameters that are common to all authentication services:

- `$service_name`: This must be set the unique name of the External Authentication Service a described in [Services](#) (page 60).
- `$reg_server_name`: Set this parameter to the name of your Registration Server.
- `$provider_code`: This is the Provider code of the Provider associated with this External Authentication Service.

- `$allowed_origins`: This setting is a list of URLs, that are the permitted origins (or “referrers”) for calls to the External Authentication Service.

The TeamDrive Agent UI or Web Portal, when calling an External Authentication Service must specify a “referrer” URL, which is checked against this list. If it is not in the list, an error will occur. After login, the browser is redirected back to the referer page.

Note that if a referrer URL is not provide, but is required, then the first URL in the `$allowed_origins` list will used, and after login the browser will be redirected back to this URL.

- `$webportal_domain`: This setting is deprecated, and is no longer used. When upgrading to the latest version of the LDAP authentication service, copy the value of this variable to the position of the first URL in `$allowed_origins` (see above) array.
- `$user_secret_salt`: This random sequence of characters **must be unique** for each installation. Once set, this value **may never change**. It is also important that this value remains secret at all times as it is used to generate the, so-called, “User Secret” value, which is used to encrypt the user’s Key Repository stored on the Registration Server.

On installation, leave this value blank, as found in the `*_config.php.example` file. The value will then be automatically set to a 54 character random sequence when first used. On upgrade, ensure that the value from the previous version is preserved.

Changing the value will result in users not being able to access their Key Repositories, which means that the user will not have access to their Spaces after a new TeamDrive installation.

However, access can be restored for the new device if the user has an old device and “Force Re-login” is requested from the Registration Server Admin Console.

- `$prev_user_secret_ver`: This variable need only be set when upgrading from a previous version of the External Authentication Service, that used an older method to generate the User Secret (see above).

---

**Note:** If you are upgrading from a version prior to version 4.5.1 (12 April 2020) then this variable **must be set appropriately** in order to ensure migration does not disrupt user access to the Registration Server Key Repository.

---

See *Upgrading External Authentication Services* (page 16) for further details on how to set `$prev_user_secret_ver`.

- `$token_encryption_key`: This random sequence of characters is used as a key to encrypt the authentication token sent to the client. The value **must be unique** for every installation.

On installation, leave this value blank, as found in the `*_config.php.example` file. The value will then be automatically set to a 54 character random sequence when first used.

This string may be changed at any time since authentication tokens are only valid of a short time.

- `$enable_debug`: Places links in the login page which allow you to retry login and provide a link to test the authentication token generated.

## 5.1.4 Authentication Service Customisation

You may change the user interface of the authentication services provided by TeamDrive by modifying the layout and content of the following files:

- `/authservice/*/index.html`: This is the default page, which redirects to `*_login.php` by default.
- `/authservice/*/*_login.php`: You can change this page to present a login page that would be recognised by your users. For example, change the page to conform to you companies CI (Corporate Identity).

Note that all changes you make to other files under the `authservice` directory will be **overwritten** when you update to the latest `td-regserver-ext-auth` RPM. As a result, make sure that you make a backup of these files before performing an upgrade.

Before making any changes to these files, make a backup copy of the file. You can then use this copy to determine if changes have been made to the file in subsequent versions of the External Authentication Service. These changes can then be incorporated in your version of the files.

### 5.1.5 Example Template Authentication

The “example authentication service” implementation provided can be used as a template to create your own External Authentication Service.

The example implementation stores the login names and email address of user in a text file. A page, “`eg_register.php`” may be used to create example users for testing the implementation.

Before beginning your customisation, make a copy of the “`eg`” directory, and replace the prefix for all files, for example, change “`eg_*`” to “`myco_*`” to create an Authentication Services for “My Company”. Internal reference to pages like `<form action="eg_login.php" method="post" enctype="multipart/form-data">` must also be changed accordingly.

The functions in “`eg_functions.php`” must be customised for your system. For example, replace the implementation of `getUserByLoginName()` with code that retrieves users of your authentication system.

The file “`eg_config.php`”, contains the configuration parameters that are required by all External Authentication Services. This file is created by duplicating the “`eg_config.php.example`” file and renaming it to “`eg_config.php`”.

Configuration parameters include `$service_name` which must be set to the name of the service. The other parameters are documented in detail in the file “`eg_config.php.example`” file.

Add your service specific parameters to your own “`myco_config.php.example`” file, and set them as required in your services “`myco_config.php`” file. You should do this by replacing the “EXAMPLE PARAMETERS” section in the config file.

## 5.2 External User Registration

Note that a user that can successfully login to an External Authentication Service does not have to be “pre-registered” on a TeamDrive Registration Server. After successful login, if the user is unknown to TeamDrive, a user account is automatically created for the user. After this point the user is visible in the Registration Server Admin Console.

Externally authenticated users can, if required, be pre-registered in the Admin Console. In this case, the Registration Server will use the email address to identify the user, and set the external authentication ID (Ext Auth ID) of the user (see below).

Once the user has been assigned an Ext Auth ID the connection between the TeamDrive user and the externally authenticated user has been established and can no longer be changed.

## 5.3 External User Data

In order to support externally authenticated users, the Registration Server requires an external authentication ID (“Ext Auth ID”) and the email address of the user.

### 5.3.1 Ext Auth ID

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

The Ext Auth ID may be any character sequence up to 100 characters in length. The character sequence used as the Ext Auth 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 Ext Auth ID is that it's unique and unchanging.

Most systems do not have a problem providing a unique identifier for a user. Note that, in general, the email address of the user should not be used as the Ext Auth ID, even if it is fixed. Generally, authentication systems have a global unique user identifier which is appropriate for an Ext Auth ID.

Note that if this identifier changes at some point, the associated TeamDrive user will incur a disruption of the TeamDrive service, including failure to login and loss of access to Space Keys stored in the Registration Server Key Repository.

### 5.3.2 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 user's email address.

The user's email address may be changed in TeamDrive or in the external system. If the change is made externally to TeamDrive, then the Registration Server will only discover the change when the user logs in again. This will not happen automatically, but it is possible to force user re-login using the Registration Server Admin Console, see *Compelling Re-login* (page 15) below.

## 5.4 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 can be done in the Admin Console. The user will be required to login again on all devices where the TeamDrive client is installed.

## 5.5 Identifying Externally Authenticated Users

When an "externally authenticated" user logs in to TeamDrive, the system needs to determine which External Authentication Service to use.

If the user is already registered on a TeamDrive Registration Server, then the server will have stored a reference to the associated External Authentication Service in the user account. This is either determined by a previous login, or by the External Authentication Service associated with the Provider of the user (see `USE_AUTH_SERVICE` below).

If not already registered, the External Authentication Service is identified using the domain (for example: "team-drive.com") of the user's email address. In other words, the part of the email address following the '@' sign.

Companies can register domains that they own with TeamDrive in order to ensure that all users with associated email addresses use a particular External Authentication Service. Domains are registered using the Admin Console as described in: *Domains and Services* (page 59).

As part of the registration, the domain must be associated with an External Authentication Service that has already been setup and deployed (as described above).

Note that if the domain of a user's email address cannot be registered with TeamDrive (for example, generic email addresses like those belonging to gmail.com) then the only alternative is to "pre-register" the users, using the Admin Console. In this case, the Provider of the user must be associated with a particular External Authentication

Service as specified by the `AUTH_SERVICE_NAME` Provider setting (see [AUTH\\_SERVICE\\_NAME](#) (page 91) for more details). In addition, `USE_AUTH_SERVICE` must be set to `True` for the Provider.

If only certain users of the Provider should use external authentication then you can set `USE_AUTH_SERVICE` to `False` and enable external authentication for each user individually.

After Identifying the External Authentication Service to be used by a user the TeamDrive Client or Web Portal redirects the user to the login Web page of the authentication service.

Note, in earlier versions of the Registration Server it was necessary to manually configure client settings (see [CLIENT\\_SETTINGS](#) (page 93)) in order to use external authentication. This is no longer required, and explicit client settings of `enable-login` and `enable-web-login` can be removed.

## 5.6 Upgrading External Authentication Services

---

**Note:** Before upgrade make a backup copy of the “authservice” directory in your Apache docs folder. You may need to compare the new installation with the old in order to ensure backwards compatibility.

---

If you are running external authentication services setup prior to version 4.5.1 (12 April 2020) of the Registration Server, then “unnamed” services must be upgraded to a named authentication service.

In order to upgrade you must use the latest version of the various external authentication implementations. This code has been refactored to make upgrade easier in the future. In particular, code and customised pages have been separated so that future upgrades can be performed easily.

Begin by installing the PHP code of the authentication service, rename the configuration file: `*_config.php.example` to `*_config.php`. All changes you make to the configuration must be made to this file.

The upgrade procedure is as follows:

- *Configuration:* Copy over configuration parameters to the new configuration file. Besides the configuration parameters that are specific to the service, you must ensure that the values `$user_secret_salt` and `$token_encryption_key` are correctly copied over to the new configuration file.
- *User Secret Generation:* Depending on the previous version of the External Authentication Service in use, the `$prev_user_secret_ver` parameter may need to be set either `v1` or `v2` (see [Upgrading the User Secret Generation Method](#) (page 17) below).
- *Service Name:* Set the `$service_name` to the new name of the service. This must correspond to the name of service created on the Registration Server.

When you access the “Login URL” for the first time, then service will check that the `$reg_server_name` and `$provider_code` correspond to the actual values of the service.

You may need to set `$tdns_proxy` to make it possible for the External Authentication Service to contact the TeamDrive Name Server.

- *Customise Templates:* Customise the `*_login.php` and `*_verify.php` pages for you purposes. Be careful to maintain the `<?php ... ?>` and `<?= ... ?>` elements in the pages.

Finally you need to add the name of your service to the `PREVIOUSLY_UNNAMED_SERVICES` Provider setting (see [PREVIOUSLY\\_UNNAMED\\_SERVICES](#) (page 91) for details).

Once this is done, test the External Authentication Service in a browser before trying with a TeamDrive client. Setting the parameter `$enable_debug` to `true` can help with testing, but be sure to set this back to `false` before deployment.

Before deployment the new authentication service must be tested. In particular, you must compare the values returned after successful login with those returned by the previous External Authentication Service.

### 5.6.1 Upgrading the User Secret Generation Method

The “User Secret” is value generated by the External Authentication Service that is used by the user to access the Registration Server Key Repository. If the value changes, the user may loose access the the Space Keys in the repostory. As a result, it is important to ensure that this does not happen during the upgrade process.

Current generation is Version 3 (v3), as of Registration Server version 4.5.1. This is the most secure version of User Secret generation which uses the SHA256 HMAC algorithm. Previous versions, referred to as: Version 1 Generation (v1) and Version 2 Generation (v2) used an MD5 hashing.

If the previous External Authentication Service used either v1 or v2 generation then you must set the `$prev_user_secret_ver` parameter to ensure no disruption of the Key Repository service.

Version 1 Generation was used by prior to Registration Server version 3.6.7 (6 November 2017), and Version 2 Generation was used prior to Registration Server version 4.5.1 (12 April 2020).

If you are not sure which version was used by the previous External Authentication Service version, check the `authservice/auth/auth_functions.php` file of the previous service.

If the file contains function `get_user_secret_v3($user_id)` then Version 3 was in use. In this case no changes due to upgrade are required and the `$prev_user_secret_ver` parameter can be left as is.

If the file contains function `get_alt_user_secret($user_id)`, then Version 2 Generation was in use. Set `$prev_user_secret_ver` to v2.

If the file contains function `get_user_secret($user_id)` and none of the functions mentioned above, then Version 1 Generation was in use. In this case, set `$prev_user_secret_ver` to v1.

If you have recently (within the last few years) upgraded from Version 1 Generation to Version 2 Generation then the “safe” option is not to upgrade to Version 3 Generation. In this case set `$prev_user_secret_ver` to `v1->v2`.

If `v1->v2` (previously `v2, v1`) has been used for quite a while (several years) then it should be possible to “upgrade” to `$prev_user_secret_ver = "v2"`. This will enable User Secret v3 generation, while still upgrading from User Secret v2. Space keys encrypted wth the Version 1 User Secret will no longer be accessable, however, the upgrade to version 2 should have largely taken place after the last few years.

Here is a summary of the `$prev_user_secret_ver` values:

- " " (blank): User Secret v3 generation will be used, no upgrade performed.
- v2: User Secret v3 generation will be used, and clients that previously used v2 will be upgraded.
- v1: User Secret v3 generation will be used, and clients that previously used v1 will be upgraded.
- v1->v2: User Secret v2 generation will be used, and clients that previously used v1 will be upgraded (this setting was previously `v2, v1`).

Please call TeamDrive for support for advice when upgrade your service, if you are unsure which value to use.

After setting the `$prev_user_secret_ver` you need to verify that the new implementation of the external authentication service returns the same User Secret value as the previous version.

To do this, compare the result page after login of both versions. Significant are the values of the hidden fields in the page named `td_user_secret` and `td_alt_user_secret`. If there is no `td_alt_user_secret` value in any of the pages, then the `td_user_secret` must match for all user’s tested.

If there is a `td_alt_user_secret` value in the page returned by the new version, then this must match with the `td_user_secret` in the current implementation. In this case `td_user_secret` in the new page is the new User Secret generated by v3 generation.

If there is a match, then the TeamDrive client will be able to access the user’s Key Repository on the Registration Server. If not, and `$prev_user_secret_ver` has been set correctly, then check the `$user_secret_salt` configuration parameter. The value must be identical to the value used by the previous version.

## 5.7 Implementation Details

The following section contains details of TeamDrive External Authentication Service implementation. This information is only relevant to developers who wish to implement and debug their own authentication service for TeamDrive.

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

### 5.7.1 TeamDrive Client: Login Initiated

External authentication users must login using their email address. This is required because the domain of the email address is used to determine the External Authentication Service to be used, if the user is not already registered.

The client sends the email address to the Registration Server in the “prelogin” call.

### 5.7.2 Registration Server: “prelogin” Call

The Registration Server determines the External Authentication Service, and returns the **Login URL** of the service to the client.

### 5.7.3 TeamDrive Client: Redirect to External Authentication Service

External authentication is a Web service, which must be accessed using a Web browser. Depending on the TeamDrive client in use, different methods are used to redirect the user to the browser based login.

These are as follows:

#### Embedded Browser

TeamDrive mobile Apps and older versions of the TeamDrive Desktop App use a Web browser embedded in the Application to access the External Authentication Service. The `req=client` Search Args is added to the **Login URL** to indicate the Embedded Browser method.

On successful authentication the service returns the required data to the client App using hidden fields in the HTML result page.

#### HTTP Redirect

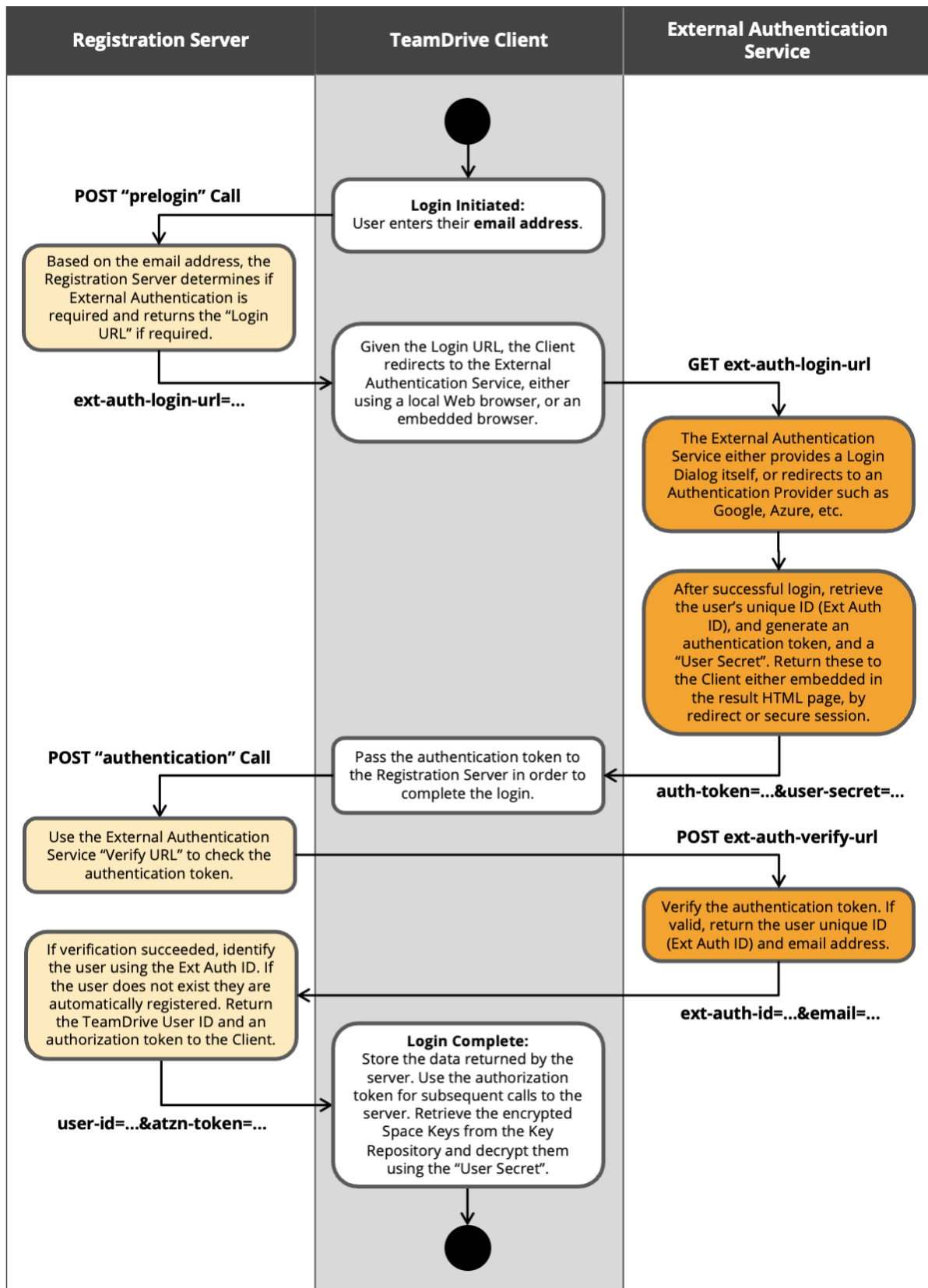
The TeamDrive Agent and the Web Portal have browser interfaces. So in this case, all that is required to access the External Authentication Service is to redirect the browser to the login page (using the **Login URL**) of the authentication service. The `req=portal&ref=<referrer-url>` Search Args are added to the URL to indicate that the HTTP Redirect method is to be used, and to specify a “referrer URL”.

If login is successful the External Authentication Service redirects back to the origin using the referrer URL provided by the client, passing data back to the client as Search Args on the URL.

#### Session Based Access

The latest version of the TeamDrive Desktop App uses “session based” browser access to perform external authentication. Authentication is initiated by launching a browser on the local machine and loading the **Login URL**.

Data is exchanged after successful login using a secure session that was established between the TeamDrive App and the External Authentication Service.



To create the session the client, first (before launching the browser) calls the **Login URL** with the Search Arg: `req=session`, and receives a JSON reply containing the Session ID (`<session-id>`) and an “Encrypted Session ID” (`<enc-session-id>`).

The Search Arg `sid=<enc-session-id>` is then added to the **Login URL** used when launching the browser. The TeamDrive App then polls the External Authentication Service using the Search Args: `req=status&sid=<session-id>`, which returns a JSON reply containing the status of the login process.

If login is successful then the `req=status` request the relevant login data to the client in a JSON reply.

### 5.7.4 External Authentication Service: Login

When directed by the TeamDrive Client, the External Authentication Service either displays the login dialog itself, or redirects to an “Authentication Provider” such as Google or Azure.

On successful login, the service returns required data to the client using one of the methods described above. The most data returned includes the following:

#### Authentication Token

The Authentication Token is an encrypted character sequence containing the data that must be returned when the Registration Server requests verification of the token. It is prefixed with the name of the External Authentication Service.

This data in the token includes the Global Unique Identifier of the user as determined by the authentication provider, and the email address of the user.

Only the External Authentication Service on which the token originates can decrypt the token. An Authentication Token has an expiry time of 2 minutes, and a CRC32 check ensure prevent corruption.

The Authentication Token is returned to the TeamDrive Client which passes the token on the the Registration Server for verification of the login:

- **Embedded Browser:** The Authentication Token is returned in a hidden HTML field called `td_authentication_token` in the result page.
- **HTTP Redirect:** The Authentication Token is returned as a Search Arg named `authToken` on the referrer URL.
- **Session Based Access:** The Authentication Token is returned in an item named `authToken` in the JSON reply.

#### User Secret

The User Secret is a hash of the Global Unique Identifier of the user as provided by the authentication provider. The hash is generated using a secret salt value (`$user_secret_salt`) known only to the External Authentication Service.

The User Secret is used by the client to decrypt and access the Registration Server Space Key Repository belonging to the user.

The latest version (Version 3) of the hashing function uses the SHA256 HMAC algorithm.

On successful login the User Secret is returned to the client as follows:

- **Embedded Browser:** In a hidden HTML field called `td_user_secret` in the result page.
- **HTTP Redirect:** As a Search Arg named `userSecret` on the referrer URL.
- **Session Based Access:** In an item named `userSecret` in the JSON reply.

## Alternative User Secret

The Alternative User Secret is optional and will only be set if the Configuration parameter `$prev_user_secret_ver` is set to a non-blank value. In particular: `v1` or `v2`.

When `$prev_user_secret_ver` is set, the Alternative User Secret contains the User Secret generated using a previously used version (either Version 1 or Version 2) of the User Secret hashing algorithm.

This provide a way to upgrade to Version 3 hashing without users loosing access to their Key Repository. The TeamDrive Client will first use the User Secret to decrypt the Key Repository and if that fails the Alternative User Secret will be used. After decryption the Space Keys are re-encrypted using the User Secret so that after a while it is no longer necessary to return the Alternative User Secret.

If required, after successful login, the Alternative User Secret is returned to the client as follows:

- **Embedded Browser:** In a hidden HTML field called `td_alt_user_secret` in the result page.
- **HTTP Redirect:** As a Search Arg named `altUserSecret` on the referrer URL.
- **Session Based Access:** In an item named `altUserSecret` in the JSON reply.

## Profile Data

In addition to the values specified above, of which the Authentication Token and User Secret are required, the External Authentication Service can also return various profile data belonging to the user.

These values are all optional and include:

- **Display Name:** The full name of the user.
- **Email:** An alternative email for the user.
- **Telephone:** The home telephone number of the user.
- **Mobile:** The mobil telephone number of the user.
- **Notes:** Generally notes regarding the user.
- **Language:** Specifies the Language used by the user.

## Deprecated Hidden fields

When using the **Embedded Browser** method, then following deprecated values are still returned in hidden fields for backwards compatibility. They will be removed in a future version of the External Authentication Service:

- **td\_login\_page=loginlostpasswordregister:** This field indicates to the embedding system the function of the page being display. This was previously used to switch the embedding panel accordingly, but is **no longer required**.
- **td\_registration\_server:** Contains the value of the `$reg_server_name` configuration parameter, and is **no longer used** by the client.
- **td\_distributor\_code:** Contains the Provider Code sent by the client when initiating the login. The value is no longer used.
- **td\_authentication\_cookie:** Contains the Base 64 encoded identifier used to login to the authentication service. This is now required to be the user's email address as registered by TeamDrive.

## 5.7.5 Registration Server: “authenticate” Call

After receiving the Authentication Token from the External Authentication Service, the TeamDrive client sends the value to the Registration Server using the “authenticate” API call.

The “authentication” Call retrieves the **Verify URL** for the External Authentication Service, and calls this URL, passing the Authentication Token.

The External Authentication Service decrypts the token, and checks that the token is valid and that it has not expired.

If all is correct then the External Authentication Service returns the data encrypted in the token to the Registration Server. This include the Global Unique Identifier of the user, also known as the “Ext Auth ID”, and the user’s email address.

The Ext Auth ID is used to find the associated TeamDrive user. If the user is found the Registration Server will update the users email address if necessary.

If the user is not found, then the Registration Server searches for a user with the given email address. If a user is found, then the server checks that the External Authentication Service associated with the user, matches the authentication service being used. If so, the server stores the Ext Auth ID in the user record.

The the user is not found, then the user is automatically registered by the server. If the email is already in use, an error will occur.

The Registration Server then generates an “Authorization Token” (known internally as the “Shadow Key”) for the user, and returns this, along with the TeamDrive User ID and other data relavent to the user such as Provider Code and Client Settings to the client.

### 5.7.6 TeamDrive Client: Login Complete

The TeamDrive Client stores the user data returned by the Registration Server. The Authorization Token is used for all subsequent calls to the Registration Server.

If Authorization Token is invalidated on the server, the client will re-initiate login.

## **SECURITY AND DATA TRANSFER**

TeamDrive combines several encryption methods to ensure total privacy for users within a Space and for their shared files. The core principle is end-to-end encryption. In other words, all data leaving a client device (endpoint) is encrypted and remains encrypted during transfer and storage. Encryption keys are generated on the endpoint and, when required, are transported securely to another endpoint.

Details of the procedures involved, and encryption methods used are described in the following sections.

### **6.1 Encryption in TeamDrive**

TeamDrive uses the following encryption mechanisms:

#### **6.1.1 AES 256 (Advanced Encryption Standard)**

The Advanced Encryption Standard is the encryption algorithm used by U.S. government agencies. It is a symmetric encryption algorithm (the same key used for both encryption and decryption) that accepts key lengths of 128, 196 or 256 bits. TeamDrive uses the 256-bit AES, with CBC block cypher chaining and PKCS7 padding.

The AES implementation uses the C-code implementation of the OpenSSL library ([www.openssl.org](http://www.openssl.org))

#### **6.1.2 RSA 3072 / 4096**

RSA is widely used public-key cryptosystems used for secure data transmission. The RSA implementation that is used is the C-code implementation provided by the OpenSSL Library ([www.openssl.org](http://www.openssl.org)).

In order to exchange data using RSA a user locally creates a matching Public and Private Key pair. The Public Key is published and made accessible to anyone wanting to communicate securely with the user. The Public Key is used to encrypt a message, which can only be decrypted by the holder of the Private Key.

TeamDrive uses RSA encryption to provide a secure messaging system which is used to exchanges AES symmetric encryption keys used to encrypt Spaces. This mechanism allows TeamDrive to securely exchange keys of any length.

TeamDrive uses 3072-bit RSA keys on the clients that are rated as secure for the next 15 years. The TeamDrive Server components use 4096-bit RSA keys. All systems have been coded to handle encryption keys of any size so that they can be increased as computing power grows in the future.

#### **6.1.3 bcrypt**

TeamDrive uses bcrypt to hash encrypt passwords on the Registration Server. bcrypt is a key derivation function for passwords based on the Blowfish cipher. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

### 6.1.4 TLS (Transport Layer Security)

TeamDrive uses HTTPS (the secure HTTP protocol based on TLS - Transport Layer Security) to communicate between client and server and for all server-to-server connections. The use of HTTPS provides security in addition to the use of the end-to-end encryption based on the RSA and AES standards used by TeamDrive.

## 6.2 Registration

When the TeamDrive Client software is installed for the first time, the user must be registered on a TeamDrive Registration Server. This requires the user enter their username (optional), an email address, and password.

An exception to this is if the user logs in using an External Authentication Service. In this case, a user account is automatically created on the TeamDrive Registration Server. The data stored in this case includes the user's email and an "Ext Auth ID" (External Authorization Identifier) which is issued by the External Authentication Service.

Registered users that install TeamDrive must login using their username or email and password. All enduser devices are registered on the Registration Server where they are associated with particular user account.

After completing initial registration, the TeamDrive Registration Server sends the user an email with an activation link. Before TeamDrive can be used, the user account must be "activated" by clicking on the link. The purpose of this is to confirm the validity of the email address provided.

TeamDrive supports two-factor authentication. When enabled, after login users will be prompted to provide additional authorization based on the type of two-factor authentication in use. This is currently either an OTP (one-time password) sent via email or a code provided by the Google Authenticator App.

### 6.2.1 The Registration Process

The registration process entails the following steps:

1. After installation is completed successfully, the TeamDrive Client retrieves the Public Key of the TeamDrive Registration Server.
2. The TeamDrive Client generates a temporary RSA key and sends the Public Key and the data entered during registration to the server encrypted using the Public Key of the Registration Server. The data includes: a username selected by the user (this is optional), the user's email address and the password. The server creates a new user account and stores the user data. The password is hashed using the bcrypt algorithm for storage purposes.
3. The Registration Server generates a new Device ID and an associated Authorization Token, which it encrypts with the temporary RSA Public Key and returns them to the TeamDrive Client in the reply. The installation is set to the "deactivated" state while activation is pending.
4. The Registration Server generates a unique activation code and sends an email to the user with a link containing the activation code.
5. The user confirms the validity of the email address by clicking on the activation link. When the link is clicked, the Registration Server sets the client installation to the "email confirmed" state.
6. The TeamDrive Client waits for the installation to be activated by polling the Registration Server. It then generates a new 3072-bit RSA Public/Private Key.
7. The TeamDrive Client sends the Device ID, Authorization Token, and the Public Key of the new RSA key to the Registration Server, in a message encrypted using server's public key.
8. The Registration Server verifies the Authorization Token and then checks that the installation is in the "email confirmed" state. If so, the server saves the public key, and sets the state of the installation to "activated". The Registration Server will not allow the client software to use the new installation until it is in the "activated" state.

After activation all messages sent from the Registration Server to the client are encrypted using the RSA public key of the client installation, stored on the Registration Server.

Requests to the Registration Server are authenticated using the Device ID and Authorization Token sent by the client. The Authorization Token is invalidated and a new token is generated whenever the user's password changes. It is also possible to invalidate the Authentication Token manually or automatically on the Registration Server which requires the user to login again on all client installations.

## **6.3 Creating a Space**

To create a Space, the user must have access to a Depot and the associated TDSV credentials document. The TDSV document contains all the information required by the TeamDrive Client to access the Host Server and create a Space in the Depot. This includes the Host Server URL, the Depot ID and an Authorization Token.

### **6.3.1 Procedure for Creating a Space**

1. When creating a Space in the TeamDrive Client, the user selects the Depot in which the Space is to be created. A default Depot is provided during user registration. After this, the user enters the name of the new Space.
2. The TeamDrive Client requests the public key of the Host Server from the TeamDrive Registration Server.
3. The client sends the username, device Public Key, User Authentication Token, the Depot ID and Authorization Token, and the name of the Space in an encrypted message to the TeamDrive Host Server.
4. The Host Server verifies the Depot ID and Authorization Token. The server also checks that the username is in the access list of users that have been authorized to access the Depot by the Registration Server.
5. The Host Server verifies the User Authentication Token by sending an encrypted message to the TeamDrive Registration Server, and retrieves the matching User ID and email address (optional) of the user's installation.

Before returning the data, the Registration Server verifies the identity of the TeamDrive Host Server (all Host Servers must be registered in the TeamDrive Registration Server Network) and encrypts the reply using Public Key of the Host Server.

6. The TeamDrive Host Server creates a new Space in the specified Depot. A unique Space ID and a 128-bit Authorization Code is generated for the new Space. The Space ID, Authorization Code and a URL used to access the Space are sent back to the client in a message encrypted with the client public key.
7. The client completes Space setup by generating a 256-bit AES key that is used to encrypt all data uploaded to the Space. The Space ID and Authorization Code are used to read and write the Space data (as explained in the section: Host Server Authentication below). These values are saved in encrypted form in the TeamDrive Client database.

## **6.4 Joining a Space (Accepting an Invitation)**

### **6.4.1 Messaging**

Joining a Space is done by using the secure messaging service provided by the Registration Server. This requires that the Registration Server act as a relay for messages. The server collects the messages and forwards them as required.

To send a message to another user, the TeamDrive Client software needs the email address or username of the enduser (recipient). For each device belonging to the recipient, the TeamDrive Client requests the user's Public Key from the Registration Server. The client then encrypts the message with this key and sends it to the Registration Server.

The Registration Server stores the messages for the TeamDrive Client on each device. All TeamDrive Clients check the Registration Server at a certain interval to see if a new message is waiting. If this is the case, the message is retrieved and deleted from the server.

Using the Public/Private Key system, only the client running on the device to which the message is addressed can decrypt the message.

### 6.4.2 The Space Invitation Document

The Space Invitation Document contains all information required by the software for connecting a user to a Space. It includes the Space URL, the Space ID, the 126-bit Authorization Code and the 256-bit AES data encryption key for the Space. To invite a new user to join a Space a message containing the invitation document is sent to each device belonging to the user. This message is encrypted with the Public Key of the target recipient.

This ensures that the message exchange is secure and only the specified invitee can view the contents of the space invitation document.

### 6.4.3 Identifying the Recipient

It is essential to ensure that invitations are sent to the correct user.

To ensure that the correct recipient receives the message, users should first contact the message recipient and request his/her TeamDrive username or current email address and then send the invitation. Only by performing an independent verification is it possible for the user to be sure he or she is not sending messages or invitations to intruders. For this reason, TeamDrive supports the additional password encryption of invitations. This requires that the sender of the invitation enter a password during the invitation process. The password is hashed using MD5 to produce a 128-bit key which is then used to encrypt the invitation using AES.

When this is done, the recipient must enter the password in order to accept the invitation. This means that the password must be sent from the inviter to the invitee via a different communication channel, for example, in person, by telephone or using an instant messaging service that is considered sufficiently secure.

### 6.4.4 Inviting Unregistered Users

In order to use the system easily and universally, TeamDrive provides a service that allows a user to invite someone who is not yet registered as a TeamDrive user.

This is done using the following procedure:

1. The user selects the Space and enters the email address of the user whom he or she wants to invite. In this case, it is strongly recommended to use a password-encrypted invitation, as described above.
2. The TeamDrive Client sends an encrypted message to the Registration Server, requesting the username of the user with the given email. The Registration Server indicates that a user with the specified email address does not exist.
3. The TeamDrive Client then creates a Space Invitation Document and encrypts this with the Registration Server's Public Key (and the password from (1) if specified) and sends this to the Registration Server, along with the email address of the unregistered user. The invitation is saved by the Registration Server.
4. The Registration Server then sends a normal unencrypted email to the unregistered user. This email message explains to the invited user how to download and install the TeamDrive Client software. The message may also include a personal message from the inviter explaining to the recipient why he or she should install the software, as well as how to obtain the additional password.
5. Once the unregistered user registers using the email address with which they were invited, the Registration Server sends the invitation document to the TeamDrive Client of the user.
6. If the invitation is not retrieved by the unregistered user within a certain amount of time (usually a number of days), the invitation is deleted from the Registration Server.

## 6.5 Accessing a Space

To obtain access to a Space, TeamDrive requires a Space URL, the Space ID, the 128-bit Authorization Code, and the 256-bit AES data encryption key. The URL contains the internet address of the Host Server. The Space ID identifies the Space, which in turn identifies the Depot in which the Space data is stored.

Changes are uploaded to- and downloaded from the Space using the HTTPS GET, PUT and POST methods. Before any file or Space meta-data leaves the client device, it is compressed and encrypted with the Space key.

### 6.5.1 Object Store Access

If the Host Server is connected to an Object Store, then the credentials for accessing the Object Store are stored in the database of the Host Server, and are not accessible to the client. Direct upload to the Object Store is not permitted. All uploaded data is first stored locally by the Host Server and transferred by a background process to the Object Store.

Direct download from the Object Store may be enabled on the Host Server. In this case, the client must first make a request to the Host Server which is authenticated as described below. The Host Server then generates a temporary read authorization for the requested data on the Object Store, and sends a re-direct to the client. The client is then able to download the data directly from the Object Store. To access the downloaded data the 256-bit AES data encryption key is still required, of course.

### 6.5.2 Host Server Authentication

Access to a Host Server is stateless. In order to send or retrieve data from the Host Server the TeamDrive Client does the following:

1. Create a URL with the required arguments, including: the Space ID, BLOB ID, read/write offset, data length and recovery ID (used to detect a server-side restore).
2. If the request is a POST then the client creates an MD5 hash of the data in the body of the request and adds this value as an argument to the URL.
3. The client adds a timestamp argument, which is the current time in seconds.
4. The client generates an authentication token by calculating the SHA-256 hash of the final URL concatenated with the Authentication Code of the Space. The resulting authentication token is then added as the last argument to the URL.

On receipt of the request, the server extracts the Space ID from the URL, and uses the Authentication Code stored in the Host Server database to verify the URL. If this is correct the server also checks that the timestamp is current.

If not the server returns an error with information about the server's current time so that the client can synchronize its clock with that of the server. Finally, if the request includes a body then the MD5 argument in the URL is verified. In cases where a request is particularly sensitive to a repeat (this includes the adding of events to a Space and the creation of a Space) the server also checks that the authentication code has not been used previously. For this purpose the server stores the authentication code of the request for a certain amount of time (long enough to ensure that the request is otherwise rejected for an incorrect timestamp).

### 6.5.3 Host Server Reply

The reply generated by the server (which is returned in the body of the HTTP reply) consists of three parts: a JSON header, a checksum and an optional binary data body.

The JSON header contains the reply parameters, including the structure of the binary data, and MD5 hash of the binary data and the Authentication Token sent with the request.

The server then generates a checksum, which is calculated using the SHA-256 hash of the JSON header concatenated with the Authentication Code of the Space.

On receipt, the TeamDrive Client confirms that the reply belongs to the request, by checking the authentication token. It then verifies the checksum by performing the same calculation as the server using the Authentication Code of the Space. The MD5 of any binary data returned is then also verified.

### 6.5.4 Space Access Security Features

The authentication mechanism described above serves a number of purposes for the security of a Space:

1. It ensures that the space cannot be accessed without a valid Authentication Code.
2. It prevents data or any part of the request from being modified while in transit, either to or from the server.
3. The client is able to verify that a reply belongs to the request sent.
4. It ensures that requests are not repeatable in cases where a repeated request can be damaging to the integrity of a Space or result in DOS (Denial-of-Service) attack.

These security features, including the fact that all Space data is already encrypted before leaving the client ensure a secure and reliable connection between TeamDrive Client and the Host Server. In addition, the TLS-based HTTPS protocol is used by default between client and server adding further security.

### 6.5.5 File Publishing

TeamDrive allows users to publish files in a Space that can be downloaded using a “Publish URL” by users that are not necessarily members of the Space. The TeamDrive client publishes a file by uploading (using the HTTPS protocol – HTTP over TLS) the current version of the file to a download area on the Host Server. Before upload, the client generates a 256-bit key that is used to encrypt the file on the server.

After upload, the server replies with the Publish URL. The URL includes the Global ID of the Space file that has been published. The Host Server does not store the encryption key. Instead the TeamDrive client adds the key to the Publish URL. When the published file is downloaded (using HTTPS), the server uses the encryption key from the URL to decrypt the published file while streaming the file to the user. In this way, published files are never stored unencrypted on disk.

## 6.6 Registration Server Key Repository

The Registration Server provides a “Key Repository” in which users can securely store their Space keys. Usage of the repository is optional so it can be disabled by each user individually, or globally via a setting on the Registration Server.

The Key Repository serves as a backup for a users Space keys, and also enables immediate access to Spaces when TeamDrive is installed on a new device. If the Key Repository is disabled, a user’s Space keys are not stored centrally, which means that the new device can only gain access to Spaces by way of invitation (self-invitation in this case).

The Key Repository works as follows:

1. To initialize the Key Repository, the client software generates new 3072-bit RSA Private/Public Key pair (note that this is different to other RSA keys generated during the Registration Process, as described in the section: The Registration Process).
2. The client encrypts the Private Key with the so-called “user secret”, which is a salted SHA256 hash of the user’s password. The user secret is stored locally on the client in an encrypted form. Since the user’s password is always sent to Registration Server in a hashed form (as described in The Registration Process section), there is no way that the user secret can be obtained without access to the user’s device.
3. The RSA Public Key and the encrypted Private Key are sent to the Registration Server where they are stored in the Key Repository.

4. If the Key Repository is enabled then, whenever the user creates or joins a Space, the client software encrypts the Space key (and other access data as described in Procedure for creating a Space) with the Public Key, and uploads the encrypted data to the Registration Server where it is stored in the Key Repository.
5. When the user creates a new TeamDrive device, after login, the client software downloads the contents of the user's Key Repository and uses the password to decrypt the Private Key. After this is done, the Private Key is used to decrypt the Space keys (and other Space access data), so that the new installation has access to all the Spaces belonging to the user.
6. If the user's password changes, only the Private Key from the Key Repository is re-encrypted by the client and uploaded to the Key Repository.

The secure mechanism used by the Key Repository ensures that no-one other than the user has access to his/her Space keys. In particular, the keys are not accessible, even by those who have access to the Registration Server.

However, it is possible for a user to lose access to his/her Space keys. This can only occur when the following two conditions hold:

1. The user has forgotten his/her password, and
2. the user has lost all TeamDrive installations.

As long as a user has a previous TeamDrive installation, the Key Repository remains accessible, even if the user's password is lost.

Nevertheless, to further guarantee access to Spaces it is recommended to make a backup of the local Space key backup file that is written to the client device whenever a Space is created or joined.



## **SUPER PIN FUNCTIONALITY**

The Super PIN functionality is required for local encryption and makes it possible to recover access to a user account if the password is lost. The full Super PIN functionality is available to the TeamDrive client version 4.7.0 or later.

Without the Super PIN, a user can “reset” their password if they forget it, however this results in losing access to the space keys in the Key Repository stored on the Registration Server. This is because the keys in the Key Repository are encrypted using the user’s password. As a result, without the Super PIN, users must ensure that they have a local backup of their space keys.

If local encryption has been enabled by the Provider, then Super PIN functionality can be enabled by the user in the TeamDrive client by enabling local encryption. This is done by adding the setting `allow-local-encryption=true` to the `CLIENT_SETTINGS`.

Alternatively, the account manager can require all members of the account to activate the Super PIN, or the manager can require Local encryption for certain user installations. Once activated, the user will be prompted to export their Super PIN recovery data, and store it in a secure place.

This includes the Super PIN itself, which is a character sequence of the form:

AAAAAA-AAAAAA-AAAAAA-AAAAAA-AAAAAA-AAAAAA-AAAAAA-AAAAAA-  
AAAAAA

and a QR-Code which contains a “Recovery URL”, that can be used to generate a “Recovery Code” for the user account. If the user has the Super PIN, it can be used anywhere in place of the user’s password. Alternatively, upon accessing the Recovery URL the Registration Server will send a Recovery Code via email. The Recovery Code can then be used to login to the user’s account.

Once the Super PIN has been activated, the user can no longer change their password without first authenticating themselves by either entering their current password, the Super PIN, or a Recovery Code. As a result, if a user loses both their password, and their Super PIN recovery data, they have lost access to their user account, unless the user’s Super PIN has been stored in the Super PIN Repository (see below).

The Super PIN can be reset by an account manager but this means that the user will lose access to the space key in the Key Repository. Users will also be unable to access any installation that uses local encryption.

### **7.1 External Authentication**

If an user account uses external authentication (for example, an LDAP server or Active Directory), then the Super PIN is still used for local encryption, however, the Super PIN or Recovery Code cannot be used in place of the password in order to login.

This is also not required because the manager of the external authentication service can change the user’s password, or allow a password change without losing access to the space keys in the Key Repository on the Registration Server.

## 7.2 Account Super PIN Settings

The Super PIN settings for all accounts under your control can be changed on the edit account page in the “Security and Keys” settings box.

Here you will also see the current values for the Super PIN settings of your account, which allow you to:

- require local encryption for Web and Desktop installations,
- require members to activate the Super PIN,
- enable the Super PIN Repository,
- view the change history for Super PIN Settings.

The Super PIN status for each user can be found on the Edit User page in the “User Data” settings box. Here it is also possible to enable the Super PIN for users individually. In addition the User Devices list indicates if an installation is using local encryption.

## 7.3 Local Encryption

TeamDrive local encryption can be enabled for Web access and for Desktop installations. Mobile installations do not require TeamDrive local encryption as these installations can be encrypted and protected by the functionality provided by the mobile device if required.

Local encryption protects all security sensitive data stored by a so-called TeamDrive “endpoint”. An endpoint is wherever the data that is stored and transported securely by TeamDrive can be accessed.

In addition, local encryption also encrypts user data that is cached by the endpoint. This means that if data is accessed “virtually” as is done when using the TeamDrive Web client, or when using the FUSE virtual file system on desktop installations, then all user data remains encrypted on the endpoint.

In order to access an encrypted installation the user must login or provide other credentials when TeamDrive is started. As an alternative to login with password, on desktop installations, users can activate “Application Protection”. In this case users setup a 6 digit PIN which will be required on startup in order to unlock the installation.

## 7.4 Requiring Super PIN Activation

The Super PIN is automatically activated when the user enables local encryption of their TeamDrive client installation, or if local encryption is required for web access to spaces, and the user logs into a Web Portal.

Local encryption provides additional security by encrypting user data in a local device installation in addition to the standard TeamDrive end-to-end encryption. Local encryption requires the Super PIN because the local data is encrypted using the Super PIN. In order to provide access to a space via the browser, a Web Portal creates a virtual device (endpoint) in the form of a container for the user. If local encryption is enabled then all data in the container is encrypted, which provides additional security in the case that Web Portal is the target of a cyber attack.

Besides local encryption, activating the Super PIN provides extra protection against password loss and against losing access to space keys stored in the Registration Server Key Repository. In addition, by enabling the Super PIN Repository managers are able to help users that loose access to their user account (see below).

In order to ensure the extra security, you can require users of your account to enable the Super PIN functionality

## 7.5 Super PIN Repository

The Super PIN Repository stores the Super PIN recovery data of all users of an account.

When enabled you can use the recovery data stored in the repository to send a user of the account a “once-off” Recovery Code via email. The user can use the Recovery Code in place of a password to login to their user account.

When the Super PIN Repository is enabled users will be required to upload their recovery data. For this purpose they will be prompted to login. If they are using the Web Portal, then the recovery data will be uploaded automatically after login.

When you enable the Super PIN Repository you will be required to create a “Master Password” which must be at least 20 characters long. This password can only be changed by first disabling the Super PIN Repository, which will delete the recovery data stored in the repository.

Store the master password in a safe place, and make it only available to trusted managers of the account. In order to send a Recovery Code to a user, you will be required to enter the master password.

### **7.5.1 Recovering from Lost Password**

A manager can help users that have lost their password, if the Super PIN Repository has been activated.

In the Admin Console, go to the User Edit page, of the user that has lost their password. In the “User Data” section you will find Super PIN status of the user.

If the user’s recovery data is stored in the Super PIN Repository it will be indicated here. In this case, the “Send Recovery Code” button will be enabled.

Click this button to send the user a Recovery Code which they can be used to login, and access the Registration Server Space Key Repository. You will be required to enter then Master Password in order to do this.

If this button is not enabled, then the user’s recovery data has not been uploaded to the Super PIN repository. This may happen if the user has not logged-in to a TeamDrive client, since the Super PIN Repository was activated.

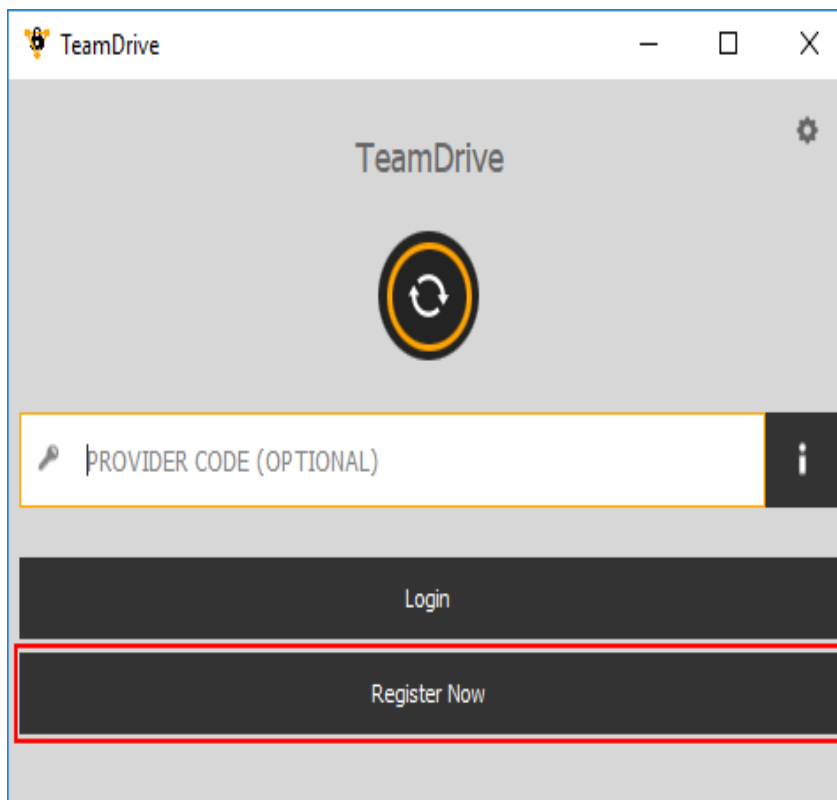


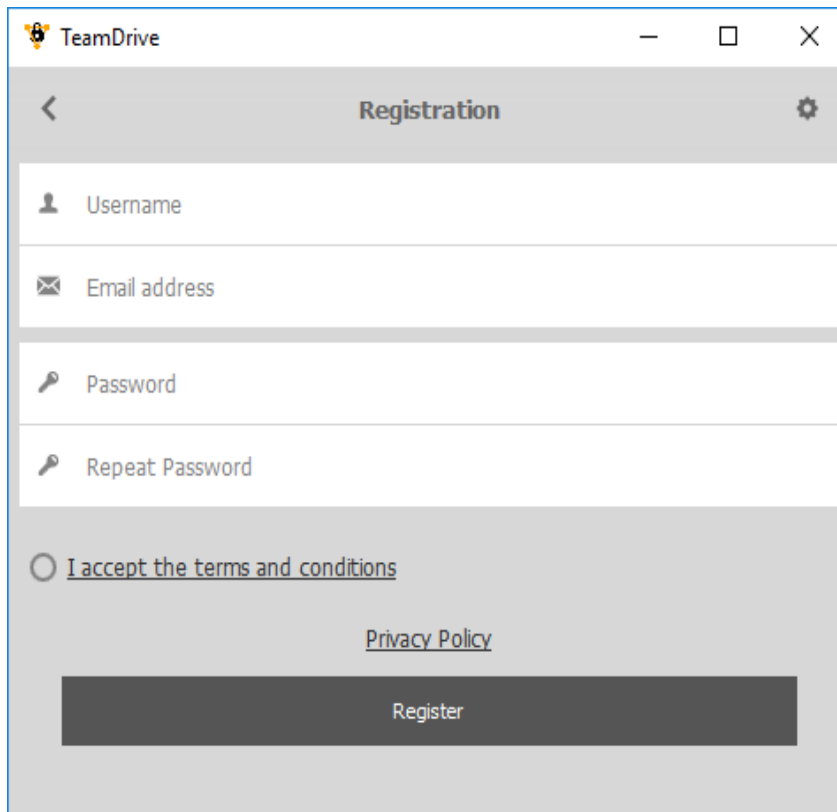
## TEAMDRIVE CLIENT-SERVER INTERACTION

### 8.1 Users

#### 8.1.1 Create a new user

A user can use the TeamDrive App to register a new user. To ensure that the user is created on the correct Registration Server and for the correct Provider, ensure that the domain of the user's email address is registered.



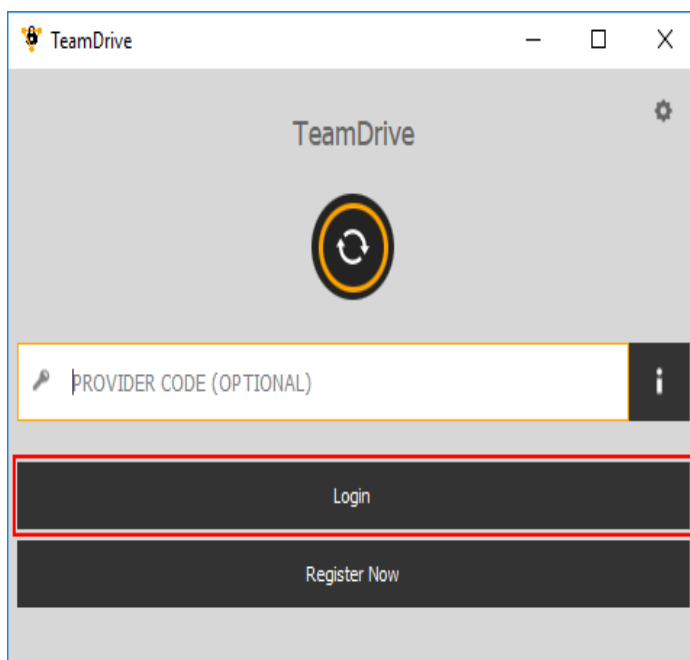
A screenshot of the TeamDrive mobile application's registration screen. The screen has a grey header with the TeamDrive logo and a back arrow. Below the header, there are four input fields: 'Username', 'Email address', 'Password', and 'Repeat Password'. Each field has a corresponding icon (person, envelope, key, and key respectively). Below the input fields, there is a radio button next to the text 'I accept the terms and conditions'. Below this, there is a link for 'Privacy Policy'. At the bottom, there is a large dark grey button labeled 'Register'.

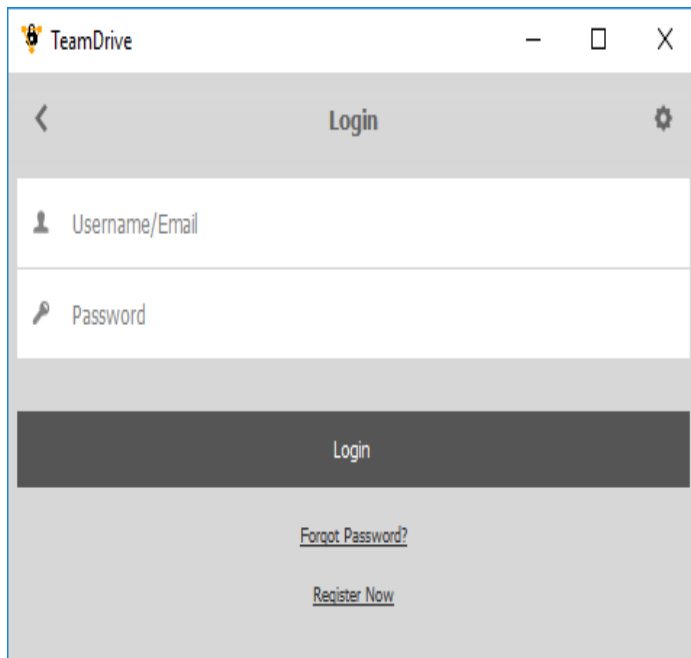
An email address and a password are required to register a user. A “username” is optional.

After registration the user account must be activated by responding to the “activation email” as described in [Email Templates](#) (page 68).

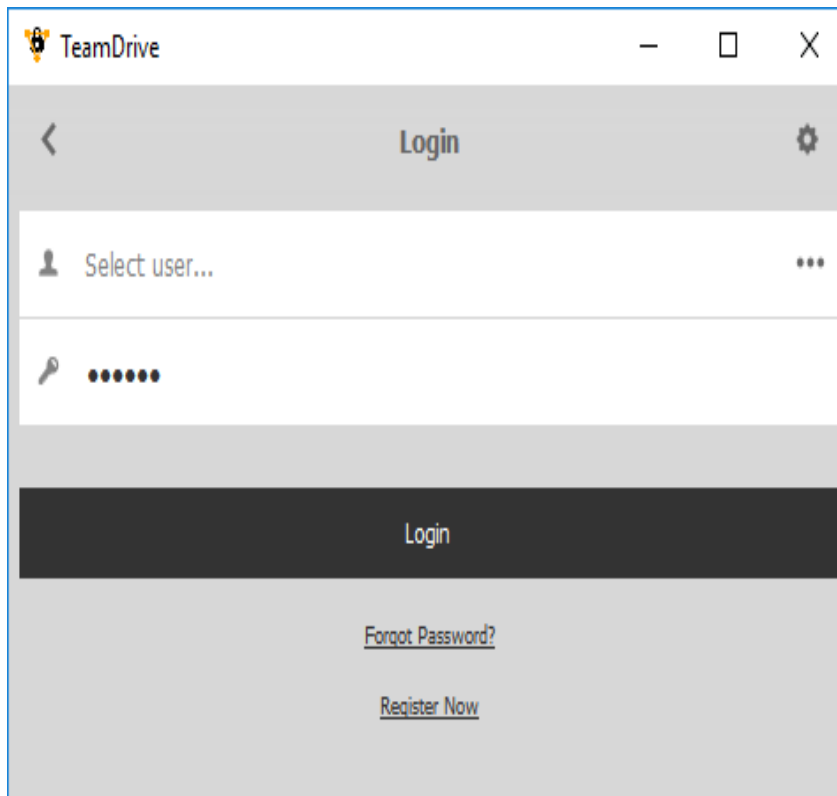
### 8.1.2 Login as an existing user

If users are already registered, they can just login without entering the Provider Code:

A screenshot of the TeamDrive mobile application's login screen. The screen has a grey header with the TeamDrive logo and a settings gear icon. Below the header, there is a large circular icon with a refresh symbol. Below this, there is a text input field labeled 'PROVIDER CODE (OPTIONAL)' with a key icon on the left and an information icon on the right. Below the input field, there are two dark grey buttons: 'Login' and 'Register Now'. The 'Login' button is highlighted with a red border.



If you enable the setting “allow-email-login” as described in *allow-email-login=true/false (default: false)* (page 113) you can also login by providing an email address. If more than one user with that email exists, you have to select the right user. Click on the . . . and a list of usernames and emails will be displayed:

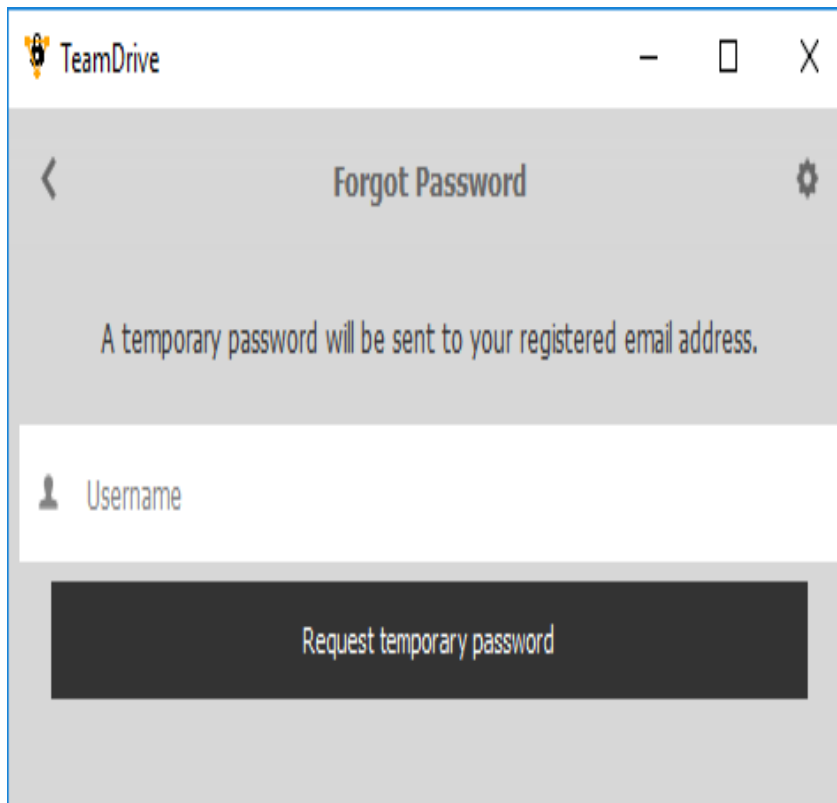


### 8.1.3 Forgotten password

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.

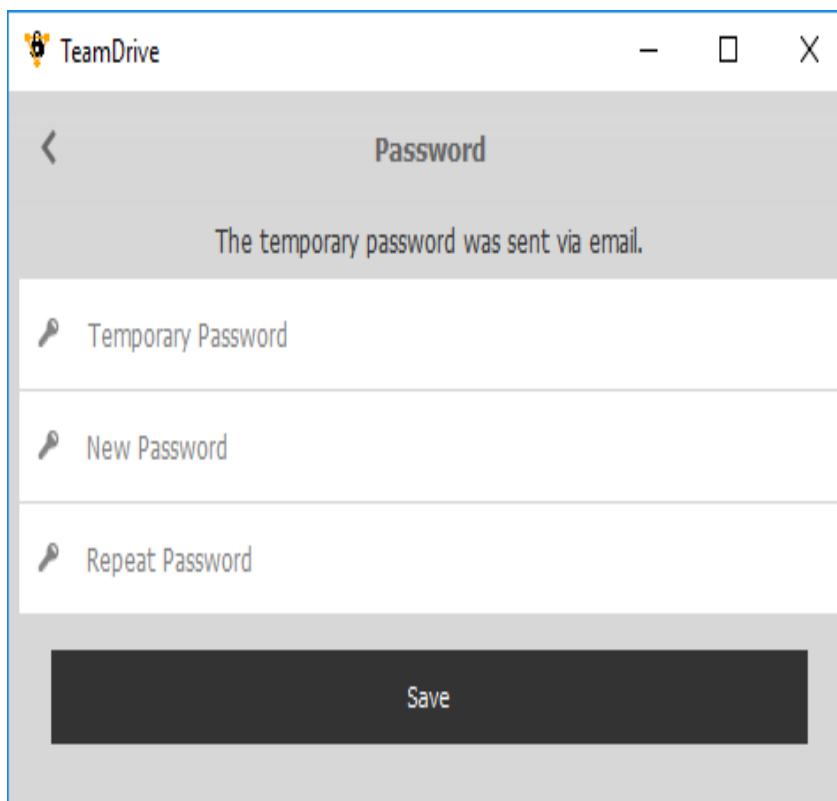
\*This can happen if a user import was performed, as described in chapter *Importing Users via CSV Files* in the

*Administration Guide.*



The screenshot shows a mobile application window titled 'TeamDrive'. The header bar is grey and contains a back arrow on the left, the text 'Forgot Password' in the center, and a gear icon on the right. Below the header, a message states: 'A temporary password will be sent to your registered email address.' Underneath this message is a white text input field with a person icon on the left and the placeholder text 'Username'. At the bottom of the screen is a large, dark grey button with the text 'Request temporary password' in white.

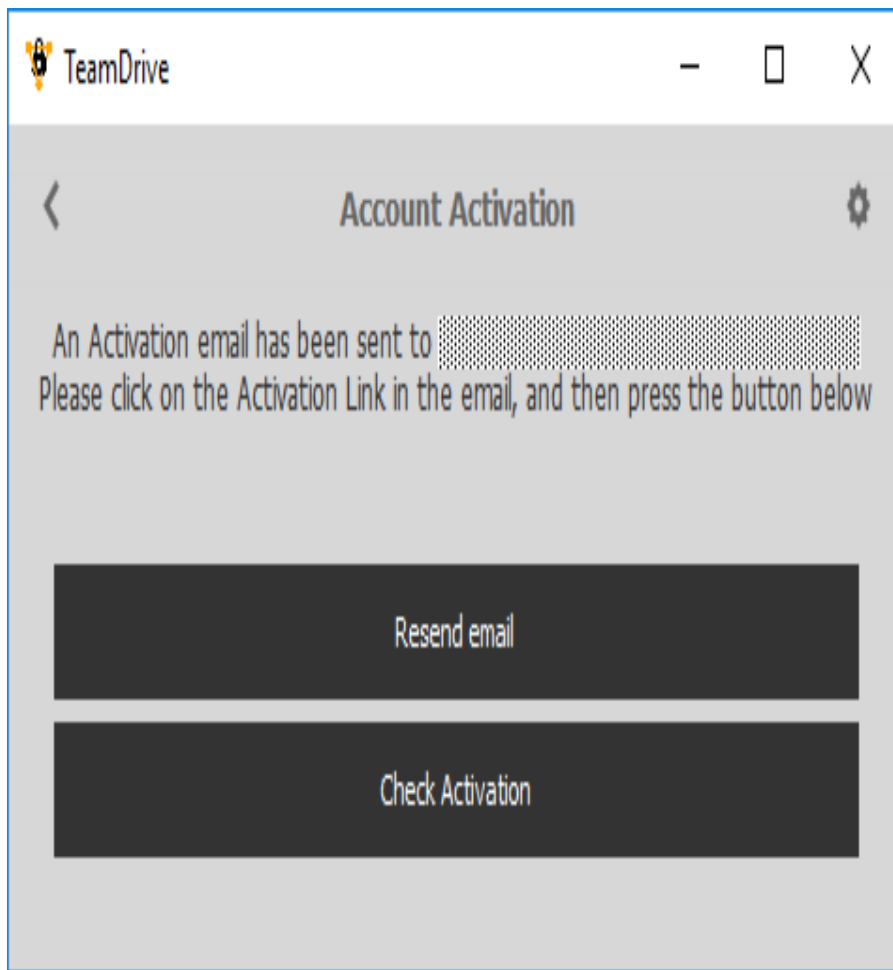
The user has to fill in his username to receive the temporary pin.



The screenshot shows a mobile application window titled 'TeamDrive'. The header bar is grey and contains a back arrow on the left, the text 'Password' in the center, and a gear icon on the right. Below the header, a message states: 'The temporary password was sent via email.' Underneath this message are three stacked white text input fields, each with a key icon on the left and placeholder text: 'Temporary Password', 'New Password', and 'Repeat Password'. At the bottom of the screen is a large, dark grey button with the text 'Save' in white.

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

### 8.1.4 Check activation



In order to finish the registration process, the user needs to click on a link in the activation email (see mail templates in *Templates for Client Actions* (page 69)). This behaviour can be modified by the settings described in *Client Settings* (page 78))

### 8.1.5 Get activation email

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

### 8.1.6 Undo registration

If the user aborts the registration process, the device (see *Devices* (page 40)) of the user will be removed.

### 8.1.7 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 102)).

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

This request asks the Registration Server for the default depot. Whether a default depot can be retrieved for the user depends on the Provider settings, see [HAS\\_DEFAULT\\_DEPOT](#) (page 98).

## 8.2 Devices

Each TeamDrive Client installation creates a new device on the Registration Server associated with the user. The user can install clients on 5 different platforms: Mac, Windows, Linux, iOS and Android OS (the number of devices per user is not limited).

Each device will create its own public-/private key. The public key is uploaded to the Registration Server for the device. When a user invites another user, a message is sent to each of the target user's devices. Each invitation is encrypted using the public key of the target device.

### 8.2.1 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 40).

### 8.2.2 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.

### 8.2.3 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 `DeviceInactiveTimeout` as described in [DeviceInactiveTimeout](#) (page 78).

## 8.3 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.

### 8.3.1 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 79)).

---

### 8.3.2 Store-forward invitation

Existing users can send out Space invitations to users that are not registered 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 registers 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.

---

### 8.3.3 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.

The TeamDrive Client generates a “User Secret” derived from the user's password. A “Global Public/Private Key” is generated on registration which is then encrypted with the User Secret. For each new Space a Space Key will be created and then encrypted using the Global User Public Key. On subsequent installations all Space Keys are retrieved from the Key Repository and the Space Keys are decrypted using the user's Global Private Key.

---

**Note:** The user's Global Public/Private Key is only used for accessing the Key Repository. The client itself uses a “Local Public/Private Key” for sending invitations to clients of other users. Accessing the user's Global Private Key only works if the user does *not change their password* during installation. A password change after login is unproblematic.

---

### 8.3.4 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.

---

### 8.3.5 Delete message

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

## 8.4 Emails

### 8.4.1 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 68).

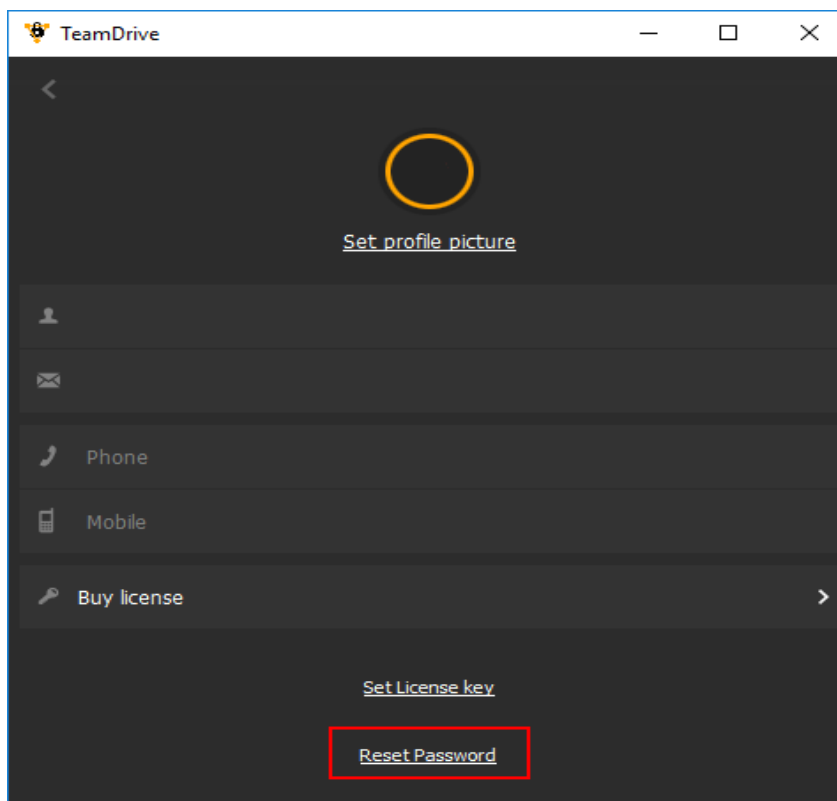
### 8.4.2 Notification email

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

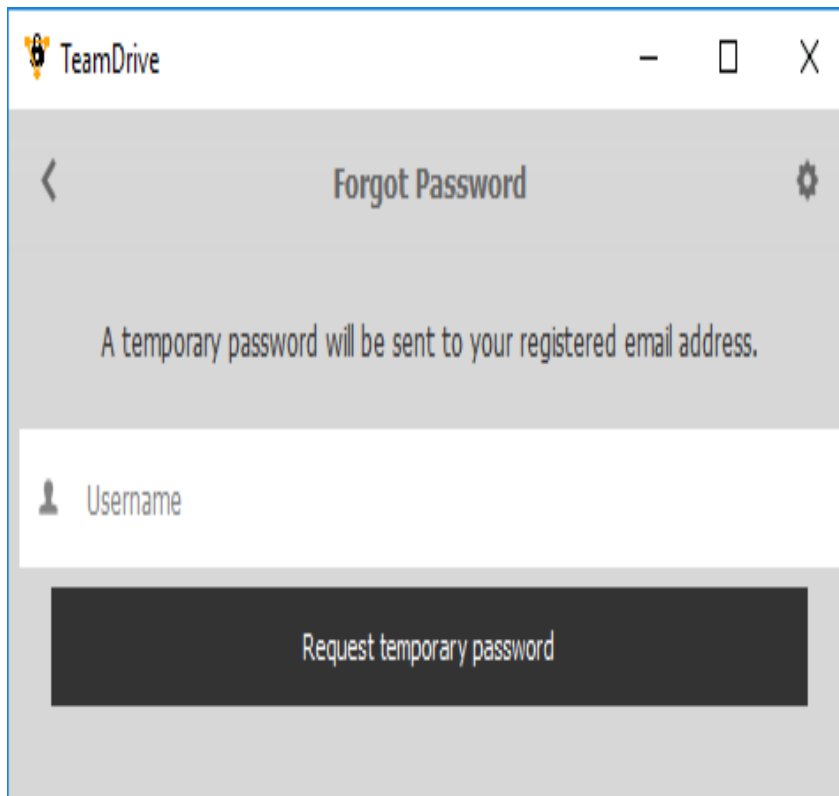
## 8.5 Change User data

### 8.5.1 Change password

The user can change their password within the client application.

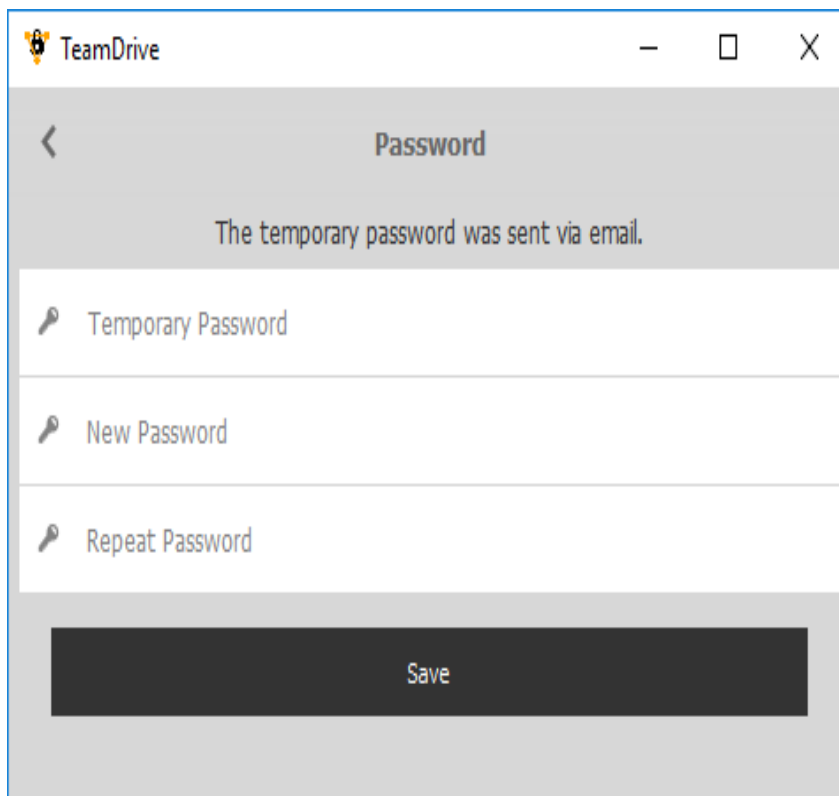


Click on `Reset Password` link on the users profile screen to get to the password change dialogue.



The screenshot shows a mobile application window titled 'TeamDrive'. The header bar is grey and contains a back arrow, the text 'Forgot Password', and a settings gear icon. Below the header, a message states: 'A temporary password will be sent to your registered email address.' There is a white input field with a person icon and the placeholder text 'Username'. Below this field is a large black button with the text 'Request temporary password' in white.

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



The screenshot shows a mobile application window titled 'TeamDrive'. The header bar is grey and contains a back arrow, the text 'Password', and a settings gear icon. Below the header, a message states: 'The temporary password was sent via email.' There are three white input fields, each with a key icon and placeholder text: 'Temporary Password', 'New Password', and 'Repeat Password'. Below these fields is a large black button with the text 'Save' in white.

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.

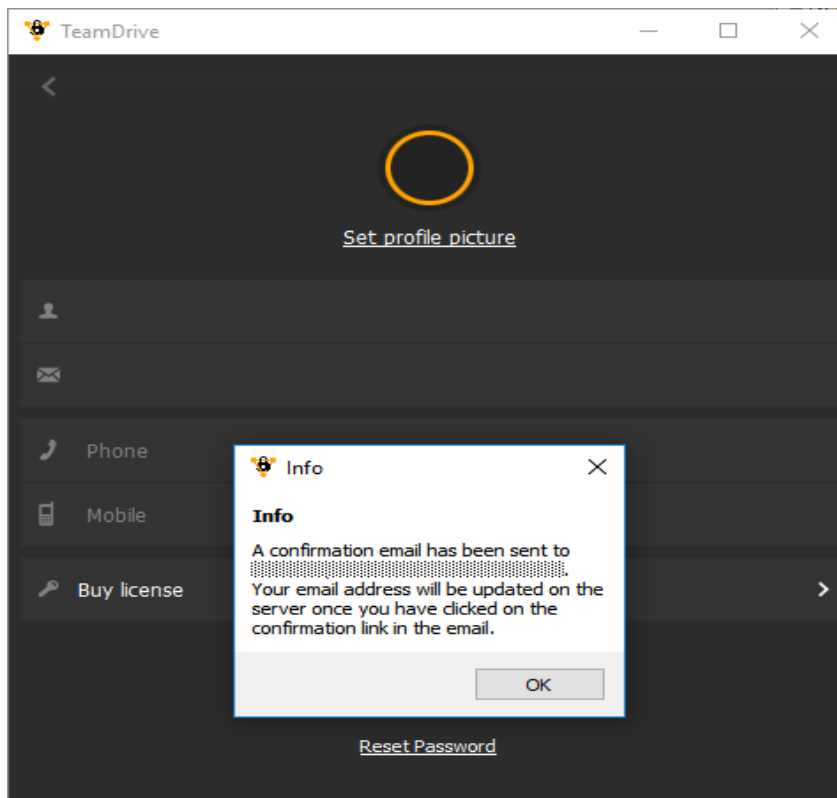
## 8.5.2 Change email

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.



## 8.5.3 License key

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

## 8.6 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*).

## 8.7 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 `update_regserver_list_task`

## 8.8 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 is provided by the API. See [registeruser](#) (page 140) for details.

---



## PROVIDER CONCEPT

A Provider is a “Tenant” of the TeamDrive Registration Server. All accounts, groups, users, licenses and Depots belong to a particular Provider. The Provider has wide-reaching administrative rights over all these object.

In addition, Host Servers, Domains and other services and endpoints are also under the control of a particular Provider.

Furthermore, most Registration Server settings can be set at the Provider level, which allows for a great deal of flexibility in the configuration of a Provider. 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-directs the user to Provider specific URLs.
- Users are directed to the Registration Server, Host Servers or External Authentication Services that belongs to, or is associated with, the Provider.

### 9.1 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.

### 9.2 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.

### 9.3 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).

### 9.3.1 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 93)). 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”.

### 9.3.2 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.

## 9.4 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 89).

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

## 9.5 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.

## 9.6 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 104)). Each license has one or more features which enable actions in the client (for more details, please look at [TeamDrive Client-Server Interaction](#) (page 35)).

If a license has an “owner” assigned (who must be an existing user of the license’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 license, see [DEFAULT\\_LICENSEKEY](#) (page 104)). 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: **WebDAV**, **Professional**, **SecureOffice**, **Agent**, **Inbox** and **Restricted** (see [DEFAULT\\_FREE\\_FEATURE](#) (page 102) for details).
- Single` or ``Multiuser license. License usage is counted per user, a single user can install and use any number of devices with one license

## 9.7 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 9)) 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.



## ACCOUNT CONCEPT

An account is used to manage a number of users. Besides users an account may have a number of resources, including licenses and depots.

Accounts belong to a Provider and have an account number which begins with the Provider Code. Account numbers are of the form:

`<provider-code>-<account-code>-9999`

`<account-code>` is a 4 letter code consisting of upper-case letters and digits, and 9999 is a random 4 digit number.

When creating an account you specify the account code, and the registration server generates the 4-digit random number, ensuring that all account numbers are different.

### 10.1 Members and Managers

A TeamDrive user may only be a member of one account. This means that existing account members must first be removed from their current account before that can be added to a different account.

An account can have a number of managers. Managers are not necessarily members of the account. Account settings, only affect the members of the account, not the managers.

Managers have the rights to manage all aspects of an account. They can create new users for the account, remove users, appoint new managers, set user licenses and manage users access to the depots used to create spaces on the TeamDrive client.

### 10.2 Using TeamDrive Shop Accounts

If you are a user of the TeamDrive shop, then you automatically have manager privileges to the account created for you, by the shop. For the most part, it is not necessary to manage your account on the TeamDrive Admin Console as the commonly used functions are provided by the shop.

### 10.3 Adding and Removing Users

Account managers may create new users for an account, but will normally not be able to add existing users to an account, as they have no access to users outside of an account. As a result, existing users are usually added to account by the Provider, which can be done using the Admin Console.

---

**Note:** The TeamDrive shop provides an option for a user to join an account. The user becomes a member of an account by entering the account number in the appropriate field. It is not possible to remove yourself as a member of an account in this manner, unless you are also a manager of the account.

---

Users created by an account manager may be assigned a license that belongs to the account, provided that the license has not reached its maximum usage limit. Alternatively a default license will be generated for the new user with the features specified by the `LICENSE/DEFAULT_ACCOUNT_FEATURE` Provider setting.

By default, this setting has the **Restricted** feature bit set, and the Provider setting `LICENSE/ACTIVE_SPACES_LIMIT` is set to 1. This means that account members using the default license only have access to one space at any given time.

When a user is added to an account as a member, the features of user's default license are changed to the features specified by the `DEFAULT_ACCOUNT_FEATURE` setting, as long as the default license of the user was not modified (in other words the feature bits of the license are set to the `DEFAULT_FREE_FEATURE` Provider setting value). Conversely, when a user is removed from an account, and the user's default license is standard for accounts, then the features of the default license are set to those specified by `DEFAULT_FREE_FEATURE`.

As mentioned above, users of the TeamDrive shop, you may elect to join an existing account. Note that, if you do this, and you are using a default license, then your license will also become restricted. As described above, when you are removed from the account, your default license reverts to the features you had before joining the account.

Users may not remove themselves from an account. This may only be done by the account manager or the Provider that the account belongs to.

## 10.4 Account Licenses

Licenses that belong to an account can be assigned to users of the account by the manager. Managers can also remove a license from a user. In this case the user will be automatically given a default license, which is created using the features specified by the `LICENSE/DEFAULT_ACCOUNT_FEATURE` Provider setting (see above).

The standard account managers privileges do not allow licenses to be created. In addition, a manager can only add licenses to an account that the user has access to. As a result, existing licenses are normally added to an account by the account's Provider.

If the Admin Console is connected to a shop (as described here: [SHOP Settings](#) (page 110)), and the user has the `PURCHASE-LICENSE` right, then a license can be added to an account by purchasing the license in the associated shop.

## 10.5 Account Depots

Depots that belong to an account can be assigned to users for their usage. This means that users are permitted to create spaces in these depots. It is also possible to "select" one of these depots which then preselected in the TeamDrive client as the default depot when creating a space.

The "Disable setting default depot on the client" prevents the TeamDrive client users from selecting a different depot as the default, to the one specified by the account manager. Since users generally create spaces without explicitly selected a depot other than the preselected default, then may ensure that spaces are created in the depot selected by the manager.

The only way to ensure that a user always creates spaces in a specific depot is to remove the usage of all depots besides the required depot from the user.

It is possible to select one of the account depots as the "Account depot", see [Depot](#) (page 53) below.

Further restrictions can be made on where users can create spaces by setting the "Supported Servers", in the corresponding box in the Admin Console (see [Supported Servers](#) (page 53) below).

## 10.6 Account Settings

### 10.6.1 Department

Use the department field to organise your accounts as required. This is a free field that may be used to qualify an account as needed.

### 10.6.2 Master User

An account master user is a user that is automatically invited to all spaces created or joined by the user's of an account. If the master user is run by a TeamDrive agent, then you should set **Enabled auto-accept invitations** for the user (see `admin_console_user_record`).

If you specify a master user for an account the master user is automatically setup for all users of the account as described here: [registration server how tos/master user](#).

### 10.6.3 Advanced Settings

The following advanced options are available, and effect all members of an account:

- **Disable network volumes** prevents users from creating spaces on network volumes.
- **Disable the Key Repository** disables the Registration Server key repository for all users in the account. Note that users that are not using the key repository need to explicitly invite themselves to spaces when they install a new device. They also have to manually backup their space keys backup file which is located in the `SpacesBackups` folder. Without this file the user cannot rejoin his spaces.

### 10.6.4 Depot

One of the depots of an account may be specified as the designated “Account depot”. This depot is made available to all account users, and is also marked as the selected depot. In this case, the depot selected for each user will be overridden by this setting.

Optionally the manager may: **Disable setting default depot on the client** to prevent users from permanently changing the default depot on client devices. The client-side default depot, is the depot that is used to create spaces if no other depot is explicitly selected.

### 10.6.5 Supported Servers

At the Account level a number of options are provided to allow the account manager to control on which systems account members may create spaces.

- **Disable import of hosting services** prevents users from importing depot access information from other sources (for example a Hosting Server not directly associated with the Registration Server).
- **Disable TeamDrive Hosting Services** prevents users from using all Hosting Services, including those managed by the Registration Server. In this case, the user's must create spaces in a TeamDrive Personal Server or a WebDAV-based service.
- **Disable TeamDrive Personal Server usage** prevents TeamDrive client users from adding access details, and creating spaces on a TeamDrive Personal Server.
- **Disable WebDAV Server usage** prevents users from adding credentials and using a WebDAV Server to create spaces.

### 10.6.6 Inbox

An inbox can be configured for an account. The inbox can be hosted by a stand alone TeamDrive Agent or using the Inbox Service hosted by the WebPortal (version 2.0.1 required). In both cases create an own TeamDrive user and assigne the user a license with the **inbox** feature.

For a stand alone TeamDrive Agent specify the following:

- **Inbox user** is a user created specifically for the purpose of importing data via the inbox.
- **Inbox Agent URL** this the URL of a TeamDrive agent running under the **Inbox user** username. The TeamDrive agent must have a fixed IP number or domain name that is accessable by all users that will be using the inbox.

For using the Inbox Service: Login with the user credentials for creating the Inbox Service hosted by the WebPortal. The Admin-Console will setup the inbox on the WebPortal server.

You can customise the inbox page using the following options:

- **Inbox banner** is an image that will be placed at the top of the inbox upload page.
- **Inbox footer** is HTML or text content that will be placed at the foot of the inbox upload page.

See `admin_console_edit_account` for more details.

### 10.6.7 Download page for published files

It is possible to customise the page used for downloading published files. The customisation affects all spaces in all the depots belonging to the account.

The following options are available:

- **Public page banner:** is an image that will be placed at the top of the published file download page.
- **Public page footer** is HTML or text content that will be placed at the foot of the published file download page.

## GROUP CONCEPT

Groups are used to control certain aspects of a user or to create teams of users that are somehow associated with each other.

Groups are administrated by a Group Manager who is usually the creator of group. User's join a group by invitation sent by the manager.

Depending on the type of user (see below), users grant control of certain aspects of their user account to the Group Manager. In particular, if a user is a member of a group then the Group Manager determines the license and the depot to be used by the group members. It is also possible to set group specific Client Settings which override the Provider level Client Settings (see *CLIENT\_SETTINGS* (page 93)) for members of the group.

But, the Group Manager cannot change any other aspects of the user such as the profile data (email address, profile picture, full name, etc.). The manager can also not access the user's devices, Spaces or the Key Repository, or delete any of these items.

However, the Group Manager can determine a Master User for members of the group (see registration server how tos/master user), which grants the manager access to all new Spaces of the group members.

### 11.1 Members and Friends

User can either belong to a group as a member or a "friend". As mentioned above, members of a group use the Group License and Group Depot when specified by the Group Manager. Members are also effected by Client Settings set for the group.

Friends, on the other hand, are not affected by any settings made by the manager of the group. Friends just have access to the membership list of a group in the TeamDrive Client. This allows them to invite all members (or all members and friends) of a group to a Space.

Users may only be a member of one group. This is to ensure that there is no ambiguity with regard to the license, depot and Client Settings used. However, a user may be a friend of any number of groups.

### 11.2 Joining a Group

Users join a group by invitation. The invitation must be sent by the Group Manager. The invitation determines whether the user will become a member or a friend of the group on acceptance.

On invitation, users receive an email with two links. With the first link they can accept the invitation, and with the second link they can reject the invitation.

In the member invitation email, the user is warned about the consequences of joining a group, namely: that by joining the group, the user gives up control of certain aspects to the Group Manager.

If the user rejects the invitation, this is noted by the server. The user can be invited again, but after the user rejects an invitation 3 times, further invitation is not possible. Users that reject an invitation are not removed from the group. They remain associated with the group, but have the `membership-rejected` or

`friendship-rejected` states (all possible states are described here: [getuserdata](#) (page 135)). Such users can be removed from the group by the Group Manager.

If a user that is already a member of a group, joins a group, the user is automatically removed from the first group. This is to ensure that the user is only a member of one group. In this case, the user gains the `invited-as-member` state in the previous group, which makes it possible for the user to return to membership of the previous group, if the user still has the original invitation email.

If a user that is a member of a group accepts an invitation as a friend then the user becomes a friend of the group, and loses membership. Conversely, a friend of a group can be invited to become a member of a group.

### 11.3 Leaving a Group

It is not possible for a user to leave a group of his own accord. A user can only be removed from a group by the Group Manager.

The group membership is noted under the user profile information in the TeamDrive Client. Here the user may also find contact information for the Group Manager.

### 11.4 Type of Groups

There are two types of groups: Provider Level Groups and User Level Groups.

Provider Groups are managed by the Provider using the Admin Console.

User Groups are managed by the Group Manager in the Group Admin Portal or by the Provider in the Admin Console.

### 11.5 Group Licenses

The Group Manager can assign a license to a group. The license must belong to the Group Manager.

When this is done, all members of the group will use the group license in place of their own license. There is no way for a group member to avoid using the group assigned license.

The license usage applies to all users of the group in a membership state. This includes: `member`, `invited-as-member` and `membership-rejected`.

In other words, if a user is invited as a member, then the user occupies a license of the group until he/she is removed from the group. Alternatively, if an invitation is still pending or has been rejected then it can be changed to an invitation as a friend and, in this case, the license of the group will no longer be in use.

In general, users that are just friends of a group do not count towards Group License usage.

### 11.6 Group Depots

A manager can assign a depot to a group. The depot must belong to the manager. When this is done, the depot becomes the default cloud storage of the group members.

Only users that have accepted membership receive access to the depot. In other words, invited users and users that have rejected membership, or friends of the group do not gain access to the Group Depot.

When a member leaves the group, or becomes a friend of the group, access to the Group Depot will be removed from the user.

## 11.7 Group Client Settings

Client Settings allow the manager to change the behaviour of the TeamDrive Client in various ways. For example, the “auto-invite-users” setting contains a list of users that are automatically invited to all Spaces.

Client Settings can be specified for all users of a Provider by setting the `CLIENT_SETTINGS` setting (*CLIENT\_SETTINGS* (page 93)). Groups present another level of control over the TeamDrive Client Settings. Client settings set for the group take priority over the Provider level settings.

Which Client Settings may be changed is controlled by the Group Admin Portal. Unlike a Provider, Group Managers are not able to apply Client Settings in a free form field.

Client Settings for a group only affect members of the group. Group friends ignore the Client Settings of the group. Of course, the Provider level Client Settings still apply to these users.

## 11.8 Group Templates

The Group system uses a number of templates that need to be customised and translated into other languages according to your requirements.

Currently, all templates are associated with the invitation of users to groups.

### 11.8.1 Email Templates

- `group-member-invitation:`

This email is sent to users that are invited to join a group as a member. The email contains 2 links: one to accept the invitation, and one to reject the invitation.

- `group-friend-invitation:`

This email is sent to users that are invited to join a group as a friend. The email contains 2 links: one to accept the invitation, and one to reject the invitation.

### 11.8.2 HTML Templates

The HTML templates provide responses to the links in the invitation email.

- `group-joined:`

This HTML page is displayed when the user clicks on the link in the invitation email to join the group. The page is also displayed if the user clicks on the reject link, but the user is already a member of the group.

- `group-notfound:`

This HTML page is displayed if the user clicks on a link in the invitation email but the link cannot be recognised for some reason. This will usually only happen if the link is incorrectly copied to the browser.

- `group-rejected:`

This HTML page is displayed when the user clicks on the link the invitation email to reject the group. The page is also displayed if the user clicks in the link to join a group, but the user has already rejected membership of the group.

## 11.9 Group Related API Functions

- `creategroup:` Use this call to create a group.
- `deletegroup:` Delete an existing group.

- `inviteusertogroup`: Send an invitation to a specific group via email.
- `removeuserfromgroup`: Cancel an invitation, or remove a user from a group.
- `setgrouplicense`: Set the license of a group. The license must belong to the Group Manager.
- `removegrouplicense`: Remove a license from a group.
- `setgroupdepot`: Set the depot of a group. The depot must belong to the Group Manager.
- `removegroupdepot`: Remove a depot from a group.
- `userjoinedgroup`: After a user has been invited to a group, this call confirms membership in the group. It performs the same functions as clicking on the accept link in the invitation email.
- `setgroupclientsettings`: Set the Client Settings of a group.
- `getgroupdata`: This call returns all information related to a group, including a list of members and their states.

## DOMAINS AND SERVICES

As of Registration Server version 4.5.1 it is possible to reserve email domains and specify named external authentication services.

As of version 4.7 all external systems associated with the Registration Server must be registered as services. This includes: External Authentication Services, Web Portals, Shop systems and other “endpoints” that access the Registration Server API.

The information is stored globally on the TeamDrive Name Server (TDNS), and requires a Registration Server connected to TDNS in order to manage the domains and services.

Domains and Services can be managed by providers using the Admin Console. Registered domains and services belong to a Provider, however, a domain can be assigned with an account (see below).

### 12.1 Domains

Domains are valid internet web addresses. In an email address it is the part following the “@” sign.

The purpose of registering a domain is to reserve the domain for a particular Provider or account. Once a domain has been registered and activated only users of the associated Provider (or account) can use email address with the domain.

By reserving their own domains, companies can control the usage of company email addresses with regard to TeamDrive. Reserving a domain, ensures that users of these email address are managed by a particular Registration Server (TeamDrive is a distributed system consisting of enterprise customer and TeamDrive hosted registration servers), Provider and Account as required.

In particular, reserving a domain, prevents a company email address from being used for a personal TeamDrive account. Conversely, this ensures that companies are able to manage any TeamDrive user’s that are registered with a company email address.

#### 12.1.1 Domain Activation

Providers can register new domains, however, they can only be activated by TeamDrive support. Please contact TeamDrive support to have the domains you have registered activate. TeamDrive support will only activate domains that actually belong to the requesting user. In particular, the domains of email service providers, such as “gmail.com”, “aol.com”, “gmx.de”, etc. cannot be reserved, of course.

If you are an account manager, then you can request a domain be reserved from you Provider (or Registration Server manager), who will then request activation by TeamDrive.

#### 12.1.2 Registration using a reserved Email Address

A reserved email address is an email address with a reserved domain.

As before, a user can be registered using the TeamDrive client, or the Admin Console. On registration an email and optional username must be provided.

A manager on the Admin Console may not create a user with a reserved email address unless they are a manager of the account or Provider that owns the domain. This ensures that users registered on the Admin Console that use a reserved email address will be associated with the correct Provider and account.

Changing the email of a user is subject to the same restriction.

When registering with a reserved email, using the TeamDrive client, users no longer need to enter a Provider code. This means that they can use the regular TeamDrive client (instead of the Enterprise TeamDrive client which requires a Provider code be entered) in order to register.

In the current release version of the TeamDrive client, on initial startup, the program requires users to enter their email address (if the user is already registered they can enter their username instead). If the client recognises the email as a reserved email it automatically directs the TeamDrive client to the required Registration Server.

Registration then proceeds as normal, and the user is added to the associated Provider and account if necessary. The license and default depot of the new user are then determined by the policies specified by the Provider and/or the account manager.

### 12.1.3 Domains and Authentication Services

Domains can be associated with an Authentication Service (see below). In this case users of the domain are required to use the specified external authentication service.

This authentication service of a domain takes priority over the external authentication service that is specified for a Provider, if any (see `AUTH_SERVICE_NAME` setting).

Usage of such an authentication service associated with a users email domain also does not depend on the value of the `USE_AUTH_SERVICE` setting of the user's Provider.

## 12.2 Services

Services provide a way to manage all external systems associated with the Registration Server. There are currently 4 types of servers: "Authentication", "Shop", "Web Portal" and "Endpoint".

The names of the services must be globally unique for then entire TeamDrive network. This is ensured by the fact that the details of services is stored on TDNS (the TeamDrive Name Server).

Services can be associated with domains, in the case of Authentication Services, as described above, and they can be associated with Providers using one of the following Provider settings: `AUTH_SERVICE_NAME`, `SHOP_SERVICE_NAME` and `WEBPORTAL_SERVICE_NAME`. These settings indicate the default service of that type used by the Provider.

### 12.2.1 Service Parameters

When you create a service you must specify the following service parameters:

- **Service name:** the unique name of the service.
- **Type:** the service type, see below.
- **Login URL:** this is the "landing page" of the service. This parameter is required for all services.
- **Verify URL:** this is required by Authentication Services (see below).
- **Authorisation:** this specifies the "authorisation type" that must be used by the service when accessing the Registration Server API.
- **IP Address List:** the is the IP Address of the service, or a list of IP addresses used by the service. This is required by all services that use the Registration Server API.

## Service Type

A service can be one of the following types:

- **Authentication:** an External Authentication Service (see below for more details).
- **Endpoint:** an unspecified service that accesses the Registration Server API.
- **Shop:** the service is a Web Shop which is used to purchase TeamDrive resources such as Licenses and Depots or for purchasing additions to these resources. For example, in order to increase user or storage limits.
- **Web Portal:** the service is a Web Portal. Web Portals are used for Web-based access to TeamDrive and for providing “Inbox” services.

## Authentication Services

TeamDrive provides standard external authentication implementations for: LDAP, Azure, Google, LDAP, Microsoft Activate Directory, Vasco IDENTIKY Authentication and standard OAuth 2.0 authentication.

In addition to the **Login URL**, all authentication services must have an associated **Verify URL**. See [External Authentication](#) (page 11) for more details.

The **Authorisation** and **IP Address List** are not required for authentication service, because these services do not access the Registration Server API.

## API Access

Services that use the Registration Server API must set the **Authorisation** and **IP Address List** parameters.

**Authorisation** can be one of the following: **MD5 (APIChecksumSalt)**, **MD5 (Endpoint specific key)** or **HMAC-SHA1 (Endpoint specific key)**.

**MD5 (APIChecksumSalt)** is the default. This is the “classic” authentication method used by all API calls before Registration Server 4.7. The service uses the key value specified by the `APIChecksumSalt` Registration Server setting (see [APIChecksumSalt](#) (page 77) for details).

**MD5 (Endpoint specific key)** is an authentication method similar to **MD5 (APIChecksumSalt)**, but instead of using the key specified by `APIChecksumSalt` it uses a service specific key. This key is generated automatically by the Registration Server and can be viewed by the administrator in the Admin Console.

**HMAC-SHA1 (Endpoint specific key)** is similar to **MD5 (Endpoint specific key)** but uses the more secure HMAC-SHA1 hashing instead of MD5 hashing. For example, in PHP you can use the `hash_hmac('sha1', $data, $key)` function.

If you alter the **Authorisation** then the service will not be able to access the Registration Server API until the new key is used by the service or, in the case of **HMAC-SHA1 (Endpoint specific key)**, until the new hashing method is used.

See [API Basics](#) (page 121) for more details on how to access the Registration Server API.

### 12.2.2 Upgrading External Authentication Services

If you are running external authentication services setup prior to version 4.5.1 of the Registration Server, then “unnamed” services must be upgraded to a named authentication service. See [Upgrading External Authentication Services](#) (page 16) for details.



## HTML AND EMAIL TEMPLATES

### 13.1 HTML Templates

#### 13.1.1 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`

---

#### 13.1.2 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 user, and an activation mail to the new email address.

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

### 13.1.3 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 `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>]]` optionally followed by `[[ELSE:<cond-var>]]` followed by `[[ENDIF:<cond-var>]]`. `IF` you may also use `IFNOT` in place of `IF` to negate the condition.

Whether an optional section is displayed depends on the value of the “conditional variable”, indicated as `<cond-var>`, in the `IF` or `IFNOT` markup tags. If the conditional variable value is empty (either `NULL` or the empty string), then condition is evaluated as “false”, otherwise as “true”.

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.

You can also set variables using the following syntax:

```
[[SET:<variable>=<value>]]
```

Variables set in this way can be used in substitutions. Note that the order of appearance in the template is not important. Conditional sections are evaluated first, then set variables are executed, and finally substitutions take place.

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/yvva/portal/<page>.html
```

In order to use the Portal Login and Registration Pages in the TeamDrive Client you must enable external authentication by setting `ENABLE_PORTAL_PAGES` ([ENABLE\\_PORTAL\\_PAGES](#) (page 92)) to `True`.

## Substitution Variables

This is a list of variable 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 set to “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>]]` `...` `[[ELSE:<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:** Set to “true” if 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:** Set to “true” if the Google Authentication code entered is incorrect.

**INCORRECT-LOGIN:** Set to “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 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:** Set to “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.

**SETUP-2FA:** Set to “true” if clicks link to setup 2-factor authentication. This variable indicates that after login, 2-factor authentication will be enabled.

**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 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 user. 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.

### 13.1.4 Set Password Pages

These pages are used in conjunction with the registration of user via the API or using the Admin Console.

The “Set Password Pages” are reached by from a link in the email templates: **web-activationsetpassword** in the case of an API call, or **reg-activationsetpassword** in the case of the Admin Console. The **web-activationsetpassword** email is sent by the “registeruser” API call when the `<setpassword>` tag is set to `true`. Sending the **reg-activationsetpassword** email is the default option when creating a user in the Admin Console.

The link in the **web-activationsetpassword** email may reference a custom set password page, which is implemented by the Provider.

The link includes arguments that include the user’s preferred language, the Provider Code and an activation code generated by the Registration Server during registration.

These pages contain variables which are replaced by the appropriate values when the page is requested. Like the portal pages they also contain optional sections which enclosed by markup of the form `[ [ IF : <cond-var> ] ]` ... `[ [ ELSE : <cond-var> ] ]` ... `[ [ ENDIF : <cond-var> ] ]` (the `ELSE` markup tag is optional). In place of `IF` you may also use `IFNOT`, to negate the condition.

Optional sections depend on “conditional variables” (indicated as `<cond-var>`) which can either be “true” or “false” (variables that are “false” are empty).

**set-password:** This allows the user to set a password and activate the user account. The user may also select whether he/she would like to receive a newsletter or not, before activation.

Users with accounts that have already been activated will not be allowed to access this page. Instead they will be redirected to the **set-password-error** page.

The following fields are available:

`[ [ PASSWORD-REQUIRED ] ]`: Set to “true” if a password was not provided by the user, otherwise empty.

`[ [ REPEAT-INCORRECT ] ]`: Set to “true” if the verification password does not match the password entered, otherwise empty.

`[ [ INVALID-PASSWORD ] ]`: Set to “true” if the password is too short, otherwise empty.

`[ [ MIN-PASSWORD-LEN ] ]`: Contains the minimum password length.

`[ [ NEWSLETTER ] ]`: Set to “true” if the user currently accepts the newsletter.

**set-password-error:** This page is returned if there is something wrong with the activation code or if the user is already activated.

The following fields are available:

`[ [ INVALID-CODE ] ]`: Set to “true” if the activation code is missing or invalid, otherwise empty.

`[ [ ALREADY-ACTIVATED ] ]`: Set to “true” if the user is already activated, otherwise empty.

`[ [ PASSWORD-ALREADY-SET ] ]`: Set to “true” if the user has already set their password, otherwise empty.

**set-password-ok:** After successful activation, the **set-password** redirects to this page. On this page you may place links to download the client software, or a link to the online Web-portal.

## 13.2 Email Templates

The templates in the Admin Console are divided into the following groups:

- API
- LICENSE-CHANGES
- INVITATIONS
- USERS
- USER-LOGIN
- DEPOTS
- GROUPS
- UPLOAD-DOWNLOAD
- SERVER-ADMINISTRATION

The group contents are hidden by default in order to provide a better overview.

Default templates are available in English and German. The language is indicated by the last component of the file name. For example: the file name: “new-passwd-de.email” is the German language email template file of the **new-passwd** template.

Each Provider has their own set of templates, so that each Provider can use their own text and graphics in the templates. A Provider must define the available and allowed languages using the `EMAIL_ALLOWED_LANG` Provider setting (see *EMAIL Settings* (page 96)).

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 in plain text. Changes to the default templates are stored in the Registration Server database.

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

### 13.2.1 Structure of the Mail Templates

**Text Emails:** In text email templates the subject and the body of the email is divided by a double forward slash: “//”.

**HTML Emails:** The structure of these templates is a more complicated (see [http://en.wikipedia.org/wiki/MIME#Multipart\\_messages](http://en.wikipedia.org/wiki/MIME#Multipart_messages)). Email clients that cannot display HTML require a plain text component in the email. If a plain text version of the email body is not provided then the contents will be shown as empty in such a Email client.

An HTML Emails template is divided into several parts. Replace the place holders 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; delsp=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'--
```

### 13.2.2 Templates for Client Actions

The following fields are available in all or a number of email templates:

- [ **[BRAND]** ]: The product brand name, defined in the Provider-specific setting EMAIL/BRAND\_NAME. If not set or empty, the default is “TeamDrive”.
- [ **[GREETING]** ] (or [ **[FULLGREETING]** ]): The greeting form used is determined by the contents of the **greetings** email template. If the username is known, then it will be included in the greeting. If only the email address is known, then a general greeting is used.
- [ **[ADMIN-CONSOLE]** ], [ **[USER-IMPORT]** ], [ **[LOGIN-PORTAL]** ], [ **[CLIENT-CALL]** ], [ **[API-CALL]** ]: One of these is set to “true” in order to specify the system that initiated the email.
- [ **[3RD-PARTY-REG]** ]: This field is set to “true” if the origin is either ADMIN-CONSOLE or USER-IMPORT. The field means that the registration of a user was initiated by a 3rd party, rather than the user himself.
- [ **[DISCLAIMER]** ]: The disclaimer text may be set in the Admin Console, in the account of a user. A text for each email language, as specified by the EMAIL\_ALLOWED\_LANG Provider setting, must be specified. If no text is available for the email language, then this tag is removed, including the end-of-line.

These templates are sent depending on certain events or actions that take place in the TeamDrive client software, the Login Portal or the Admin Console.

**activation-pending:** This email is sent to the user when manual activation of a device is required (see [requiring\\_manual\\_activation](#)). The email is used to inform the user that manual activation of the new device is necessary to complete the installation, and that the manager of the user account has been notified.

**activation-required:** When manual activation is enabled (see [MANUAL\\_ACTIVATION\\_REQUIRED](#) (page 106)) this email is sent all users on the list specified by the NEW\_DEVICE\_NOTIFICATION\_LIST setting. See [requiring\\_manual\\_activation](#) for a description of how this feature works.

**devices-disabled:** If the number of active devices of a user differs from the value specified by the MAXIMUM\_DEVICES\_PER\_USER Provider setting (see [MAXIMUM\\_DEVICES\\_PER\\_USER](#) (page 94)) then the server will enable or disable user devices accordingly.

When this happens the server sends an email using this email template to inform the user which devices have been enabled, and which have been disabled.

[ **[DISABLED-DEVICES]** ]: This is a list of devices that have been disabled. The list includes: ID, name, platform, type, creation time (indicated by “(\*)”) and last active time (indicated by “(\*\*)”) of each device.

[ **[ENABLED-DEVICES]** ]: This is a list of devices that have been enabled. The list includes: ID, name, platform, type, creation time (indicated by “(\*)”) and last active time (indicated by “(\*\*)”) of each device.

**inv-email-invited (old name: td3-privacyinvited-email):** If a new user was invited who is currently not registered, they will get an invitation sent to their email address 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.

The following additional template variables can be used in this email template. Note that the first 2 variables differ only in the line endings used.

[ **[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 109).

**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-newuser-invited:** This template is sent instead of **inv-user-invited** when the client sends an invitation to a user, and the user is automatically registered by the Registration Server. This is done when the setting `INVITATION_CREATES_USER` is set to `True`.

If the new user uses external authentication then this template is only used if the setting `ACTIVATE_ON_INVITATION` is set to `True`.

[ [ ENCRYPTED ] ]: Is set to `true` if the invitation is encrypted with a password.

[ [ AUTO-ACTIVATE ] ]: Is `true` if setting a password will also activate the user's account.

[ [ EXTERNAL-AUTH ] ] Is `true` if the invited user uses external authentication.

**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-device-notification** This email is sent to all users on the list specified by the `NEW_DEVICE_NOTIFICATION_LIST` setting when a new TeamDrive device is installed. Note that a different email, **activation-required**, is sent if `MANUAL_ACTIVATION_REQUIRED` is `True`.

**new-passwd:** The email template is used when the user requests a temporary password in order to change their current password. Setting a new password may also be done during the login process (see [Forgotten password](#) (page 37) for details).

Note that the user password cannot be changed if the Provider setting `ALLOW_PASSWORD_CHANGE` is set to `False` (see [ALLOW\\_PASSWORD\\_CHANGE](#) (page 92)).

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.

[ [ NEWPASSWORD ] ]: This template variable is replaced by a temporary password generated by the Registration Server.

[ [ SUPERPIN ] ]: This is a conditional template variable that indicates whether the Super PIN functionality has been enabled for the user account, or not.

If the Super PIN has been activated, then it is not possible to change the password using a temporary password. Instead the user is required to use their Super PIN, or a Recovery Code obtained using their recover URL.

As a result, the conditional section in this email is used to inform the user of the Super PIN requirement, if they try to use an old TeamDrive client to change their password, after the Super PIN has been activated.

**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 sent to the user:

[ [SERVERURL] ]: This is the URL defined in the xml file as described in [RegServerURL](#) (page 84). 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 (“yvva”) 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 63)).

**reg-activationnotify:** By default, only the first installation must be manually activated (depends on the setting described in [LOGIN\\_WITHOUT\\_ACTIVATION](#) (page 106)). The user will just receive a notification mail that an additional device was installed.

**reg-activationsetpassword:** When a user created in the Admin Console the default option is to send an email using this template. This email contains a link to the **set-password** HTML template page, which allows the user to set his password, and activate his user account (see ref:[html\\_templates\\_set\\_password\\_pages](#)).

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-activationwithnewsletter:** This template is sent in place of **reg-activationlink** if the user accepted receiving the newsletter in the client. The email is used to both activate the user and to accept the receipt of the 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-registrationnotify:** This email is sent after a user has successfully set a password using a link in a **activationsetpassword** email, if the Provider setting `ACTIVATE_ON_INVITATION` is set to `True`. In this case, the template variable [ [PASSWORD-SET] ] is also set to *true*, and can be used for conditional sections.

**too-many-failed-logins:** This email is sent to user when the attempted number of logins exceeds the number specified by the `ALLOWED_LOGIN_ATTEMPTS` Provider setting (see [ALLOWED\\_LOGIN\\_ATTEMPTS](#) (page 105) for more details).

### 13.2.3 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 recieved this email.

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

### 13.2.4 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 *INVITATION Settings* (page 99).

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 user's default depot.

**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 user's default depot.

### 13.2.5 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 detects 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).

### 13.2.6 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 90).

The links within the templates must reference a page that has access to the Registration Server API.

For more information on using the Registration Server API, see *API Basics* (page 121).

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

**web-activationsetpassword:** Similar to **reg-activationsetpassword**, but the link may be changed to reference a custom page created by the Provider (see *ref:html\_templates\_set\_password\_pages*).

**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-depotchanged:** This email is sent of the user's depot configuration changed. This can be in the form of a addition or removal, or the default depot is changed on the server.

**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**.

**web-user-deleted:** This email is sent to a user to confirm deletion of the user's account.

**web-registrationnotify:** This email is sent when a user is registered using the "registeruser" API call, and the user is automatically activated. If the user is not automatically activated, then the **web-activationlink** or **web-activationwithnewsletter** email will be sent.

### 13.2.7 Mail Templates for API License Changes

In the Admin Console you can determine whether a license change email is sent to the license owner (or holder) using the checkbox titled **Send license change email**. The checkbox is set to checked by default if the `API_SEND_EMAIL` Provider setting is set to `True`.

With regards to the API, whether an email is sent to the license owner is determined by the `<sendmail>` tag which may be set to `true` or `false`. If this tag is missing then by default an email is not sent except in the case of the "resetlicensepassword" API call.

Other API calls that do not involve changing licenses use the `API_SEND_EMAIL` Provider setting as the default. This includes calls involving change of password and changes to a users depot configuration.

Note that a license change email are always sent to the Provider regardless of any settings or user selection. The providers **License email** address is used for this purpose.

If the license has no owner, then the value set for the **Holder email** of the license is used. The language used in the email can also be set in the appropriate license field for this purpose.

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

**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-license-rec:** A license confirmation mail for the holder of a newly 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: WebDAV, Professional, SecureOffice, Agent, Inbox and Restricted.

`[ [VALID-UNTIL] ]`: In case of license with an expiry date.

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

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

## 13.2.8 Mail Templates for Account

**account-configuration:** This is the template for an email which includes a summary of an account configuration. It can be sent by an account manager or administrator to all managers of an account using the Admin Console.

**account-manager-invitation:** This email template is sent to invite a user to an account as manager. This template is also used if the user is invited as both member and manager. The email contains links which allow the user to either accept or reject the invitation.

**account-member-invitation:** This email is sent to invite a user to an account as member. The email contains links which allow the user to either accept or reject the invitation.

## 13.2.9 Mail Templates for Groups

**group-friend-invitation:** This email is sent to invite a user to a group as a friend.

**group-member-invitation:** This email is sent to invite a user to a group as a member.

## 13.2.10 Mail Templates for Inboxes

**inbox-confirm-upload:** This email is sent to the user that uploaded files to an inbox, if the email address of the user is known.

**inbox-upload-notification:** This email is sent to all members of a space to indicate that files have just been uploaded to an inbox in the space.

## 13.2.11 Mail Templates for Depots

These mail templates are used by Host Server associated with the Registration Server in order to send notifications to managers and owners of depots.

The following template variables may be used in these emails:

[ [DEPOTNAME] ]: The name of the depot.

[ [DEPOTID] ]: The ID of the depot.

[ [HOSTSERVER] ]: The Host Server domain of the depot.

[ [DISKUSED] ]: The amount of disk storage currently in use by the depot. This value includes the unit: GB, MB or KB depending on the amount.

[ [DISKLIMIT] ]: The disk storage limit of the depot. This value includes the unit: GB, MB or KB depending on the amount.

**depot-warning:** This email is sent to warn the user that their depot disk storage usage has exceeded a certain threshold.

[ [USEPERCENTAGE] ]: This is the percentage that has been exceeded (for example “80”). If entire limit (100%) has been exceeded then, this value is set to empty.

**depot-reduced:** This email is sent to inform users that the spaces in the user’s depot have been deleted in order to reduce the disk storage usage to the required limit.

It is possible to undelete the spaces within the next `SpaceDeletionDelay` minutes.

[ [SPACEID-LIST] ]: A list of space ID’s of the spaces that have been deleted.

**depot-reduction:** This email is sent to inform users that the spaces in the user’s depot will be deleted (in a certain number of days) in order to reduce the disk storage usage to the required limit.

[ [DAYSREMAINING] ]: The number of days remaining before spaces are deleted.

**depot-cancelled:** This email is sent if previously a **depot-reduction** email was sent, and since then the disk usage or limit has changed so that the a removal of spaces is no longer required.

**depot-traffic:** This email is sent if the depot network traffic exceeds certain usage thresholds.



## SETTINGS

### 14.1 Registration Server Settings

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

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

#### 14.1.1 API Settings

##### APIAllowSettingDistributor

When accessing the API, providers are identified by the IP address of the caller (see [API Access](#) (page 61)).

Set this to `True` if you want a Provider to be able to make requests on behalf of another Provider. This means that a Provider that manages (see the “Managed by” setting on the Providers page in the Admin Console) other providers can set the caller Provider to one of those providers.

The “Default Provider”, specified by the `DefaultProvider` setting (see [DefaultProvider](#) (page 83)) has the right to manage all other providers.

In order to make an API request on behalf of some other providers, set the `<distributor>` tag to the required Provider Code (see [API Input Parameters](#) (page 123)).

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 121) 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`).

### RegServerAPIURL

Optional Reg Server API URL, used by the Administration Console (e.g. `http://regserver.yourdomain.com/yvva/api/api.xml`). 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`.

### WebPortalAPICalls

This is a comma separated list of API calls that are permitted for the Web Portal. If set to empty, Web Portals will not be able to access the Registration Server.

There should be no need to update this setting. New calls required by the Web Portal will be added in future updates as required.

## 14.1.2 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.

### 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 93) for details.

---

### ClientUsernameLength

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

### DeviceInactiveTimeout

If a user device is not used for the time specified by the `DeviceInactiveTimeout` setting it is considered inactive. You can re-activate a device by starting the TeamDrive client on the inactive device.

The Registration Server keeps a list of devices created by each user. Each new installation of the TeamDrive client will create a new device entry.

Invitations are not sent to inactive devices. If a user has no active devices, the TeamDrive client will report an error when the user is invited to a Space. In this case, you should contact the user who should either re-activate an old device or create a new TeamDrive installation.

Device activity is only updated once per once a day. So, the value of `DeviceInactiveTimeout` should not be less than one day (86400 seconds). The default value is 96 days (8294400 seconds).

### 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.

### 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 set to 0, invitations will never be deleted. Deletions are carried out by the background task described here: `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 41) for a detailed description.

### IPAddressStoreTime

IP Addresses stored by the Registration Server will be removed after this time. The default is 7 days. On upgrade to Registration Server 5.0.2, if the value is greater than 7 days, it will be set, once off, to 7 days.

The IP addresses stored when creating a new device is optional. If `IPAddressStoreTime` is set to -1 seconds, then the server will no longer store IP addresses when a device is created. Furthermore, if set to 0 then the IP address will not be deleted from the device record.

For security reasons, in **all other** cases of IP Storage, a minimum of 7 days storage is applied. In this case, if `IPAddressStoreTime` is set to -1 or 0 then a 7 days storage period is used, which is the maximum time permitted under European law.

Please check your local requirements under the law before changing this setting.

The “Remove IP Addresses” auto-task as described in `remove_ip_addresses_task` is responsible for removing expired IP addresses from the database.

### 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.

## 14.1.3 Email Settings

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

### EmailDNSCheck

When set to `True` the Registration Server will check the domains of registration emails using an `nslookup` system call. If the call fails the email address will be rejected. By default this value is `False`.

### EmailHookIPList

This is a list of IP addresses that are permitted to call the email notification/event hook (see `EmailHookURL` below). By default the list is empty, which means calling the email hook is not permitted.

Set to `*` to enable notifications, but require that callers to include the “hook key” in the URL as specified by `EmailHookURL`.

### EmailHookURL

This is the URL of the email hook used for notification of various email related events sent by various email services. You may use this URL without the “key component” if the IP address of the origin is specified in `EmailHookIPList`.

With the key component the URL has the form:

```
https://<reg-server-domain>/bal/emailhook.json``.
```

### EmailRetentionPeriod

The time that emails must be preserved after they have been sent, cancelled or deleted. By default, 60 days.

### EmailSendRate

This is the maximum send rate for emails per minute. The default is “0” which means unlimited.

### FailedEmailTimeout

The time that emails must be preserved after they have failed, bounced or been blacklisted. By default 180 days.

### 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. The email will also be used as the ‘envelope-from’-email in user-to-user mails like invitations and as the ‘from’-email for all server-to-user emails like the activation email, new password, etc.

---

**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 96) 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 and/or receive per day. (-1 = unlimited, 0 = no mail)

### **MaxInboxEmailPerDay**

This setting specifies the number of emails an inbox can sent per day. This on concerns the upload confirmation emails which are sent to unregistered users. The upload notification emails are always sent to registered users and are included in the `MaxEmailPerDay` which each user may receive.

As for `MaxEmailPerDay`, “-1” mean unlimited, and “0” means no email will be sent.

### **ResetEmailLimit**

The maximum number of emails that can be reset at once, other emails will be paused, so that the email server is not flooded. The default is 20.

### **SMTPServer**

The IP or DNS name of the SMTP server.

In order to use a TLS/SSL connection to the SMTP server prefix the host name of the server with “smpts” protocol, for example: “smpts://my.smtpserver.com”. If no protocol is specified then “smt” is assumed.

### **SMTPServerUser**

An username for smtp authentication.

### **SMTPServerPassword**

The password for smtp authentication.

### **SMTPServerTimeOut**

Timeout parameter in seconds for `sendmail` requests.

### **TemplatePath**

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

### **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.

### 14.1.4 Failed Lookup Control

The lookup functions are all API calls that are used during login, registration and when inviting users. We limit the number of calls to these functions to a certain rate per hour, for a given IP address.

The manager of the default Provider is sent a notification email if the rate of failure is increasing, or if a user exceeds the `FailedLookupLimit`.

#### **CalculatedLookupMaximum**

This is the maximum call rate during the last 48 hours. This value is calculated by the “Manage Failed Lookup” auto-task every 4 hours.

If this rate is exceeded an email notification will be sent to the manager of the default Provider.

#### **CheckUserLimit**

The limit (per day) to the number of users that can be checked as to whether they exist or not. This value is set to 200 by default.

#### **FailedLookupLimit**

This is the maximum number of failed searches for usernames and email addresses that are allowed in one hour. A search occurs when a user is invited to a Space, or during login and registration. By default this value is 200.

The value is intended to prevent using the Registration Server API to enumerate all registered users.

Note that this also limits the rate at which unregistered users can be invited to a space using an unknown email address.

#### **FailedLookupPeriod**

The period in which the maximum failed lookup rate is enforced by the server. By default this is 30 minutes. This means if the limit is exceeded (see `FailedLookupLimit` below), then this is the maximum time that a caller must wait before they can try again.

Decreasing the period “smooths” out the enforcement of the limit.

#### **LastLookupNotification**

The time of the last notification. Notification will not be sent faster than once every 5 minutes.

#### **LookupRetentionTime**

This is the time that entries in the failed lookup log are retained. By default this is 180 days.

#### **RecentLookupMaximum**

The “Manage Failed Lookup” auto-task resets this value to the `CalculatedLookupMaximum`. If `RecentLookupMaximum` is exceeded during normal operation by frequently failed calls to lookup functions, then an email is sent to the default Provider manager.

`RecentLookupMaximum` is then set to the new maximum lookup call fail rate, and a new email is only sent when this value is exceeded again.

In this manner the manager is informed of constantly increasing fail rate.

## 14.1.5 General Settings

### AuthorizationSequence

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

This information is uploaded to TDNS (TeamDrive Name Server), and is shared amongst trusted Registration Servers on the TeamDrive network, see *Blacklisting and Whitelisting Registration Servers* (page 10).

### AssumeHttpsAccess

If set to `True` then the Registration Server will assume that clients are using HTTPS to connect to the server. The default value is `False`.

NOTE: it is **only** necessary to set `AssumeHttpsAccess` to `True` if the automatic detection does not work for some reason.

The most likely scenario for this is if the Registration Server is behind a load balancer, and the load balancer does not set the `HTTP_X_FORWARDED_PROTO` HTTP header.

Note that if the load balancer does set the `HTTP_X_FORWARDED_PROTO` header then it is necessary to set the “yvvva” setting: `UseXForwarded=True` in the `yvva.conf` file.

### 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.

### DefaultProvider

Select the existing Provider 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 47).

### EnableSuperPINRepository

If `False` (the default) the option to enable the Super PIN Repository, and the function to require account users enable the Super PIN are not available in the Admin Console.

If set to `True` the Super PIN Account level options become available to account managers in the Admin Console. In addition, all account managers are prompted by a banner to read information about the Super PIN and the options available to accounts and users.

### 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,

### 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.

This information is stored on TDNS, and is shared amongst trusted Registration Servers on the TeamDrive network, see [Blacklisting and Whitelisting Registration Servers](#) (page 10).

### 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.

This information is uploaded to TDNS (TeamDrive Name Server), and is shared amongst trusted Registration Servers on the TeamDrive network, see [Blacklisting and Whitelisting Registration Servers](#) (page 10).

### 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.

This information is uploaded to TDNS (TeamDrive Name Server), and is shared amongst trusted Registration Servers on the TeamDrive network, see [Blacklisting and Whitelisting Registration Servers](#) (page 10).

### ServerLogFiles

Location of various server log files that can be viewed from within the Administration Console via **Admin -> 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 Adminstration Console. Please ensure that the timezone is valid (see `/usr/share/zoneinfo/` for available time zone information)! (default: Europe/Berlin)

## 14.1.6 Proxy Settings

### 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/)).

---

## 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.

## 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`).

### 14.1.7 Redirect URLs Settings

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 89)). 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`.

### DownloadURL

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

### FAQURL

An optional link to a FAQ page. This URL can be overridden by the REDIRECT\_FAQ Provider setting.

### ForumURL

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

### HelpURL

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

### 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.

### 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 `client_log_files`.

### PrivacyURL

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.

### 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.

### RedirectorProtocol

The setting applies to the portal pages, the Provider “REDIRECT” settings, the global redirect URL settings and the global “RedirectURL” setting. These are collectively known as the “Redirect URLs”.

The redirect URL's are requested by the TeamDrive client in various situations, or when the user requires additional information. For example, DownloadURL or REDIRECT\_DOWNLOAD, is requested by the TeamDrive client when it directs the user to the location of client software updates.

If RedirectorProtocol is set to “https”, then HTTPS is used for all of the redirect URLs. When set to “http” then HTTP is used, but only in the cases where HTTPS, is not explicitly specified in the URL specific setting.

This means that, if a setting such as `REDIRECT_DOWNLOAD` is set to a URL like: `http://my.server.org/download.html`, and `RedirectorProtocol` is set to “https”, then a request for `REDIRECT_DOWNLOAD` will return `https://my.server.org/download.html`.

`RedirectorProtocol` may be set to either “http” or “https”, “https” is the default.

Before version 4.6.3, the default value was blank which meant that the protocol of the URL specific setting was not changed.

This setting is new in Registration Server 4.1.3.

## ReferralURL

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

## 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.

## TutorialURL

An optional link a tutorials page. This URL can be overridden by the `REDIRECT_TUTORIALS` Provider setting.

## 14.1.8 Security Settings

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

### EnableSyslog

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

### EnableXForwardedFor

Set this value to `True` if the Admin Console should read the “X-Forwarded-For” HTTP header. This is required if the Admin Console is configured to run behind a load balancer or other network component with ssl offloading functionality.

In this case the Admin Console is not directly contacted by the user’s Web-browser, and the IP address of the browser is placed in the “X-Forwarded-For” header by the proxy.

Comment out this line in the `/etc/httpd/conf.d/td-regserver.httpd.conf`:

```
YvvaSet use-x-forwarded=true
```

and comment out the HTTP-to-HTTPS-Rewrite-Rule in `/etc/httpd/conf.d/td-regserver-adminconsole.conf`:

```
#RewriteCond %{SERVER_PORT} !^443$
#RewriteCond %{REQUEST_URI} ^/adminconsole.*
#RewriteRule ^.*$ https://%{SERVER_NAME}/adminconsole/login [L,R]
```

and comment in:

```
RewriteCond %{HTTP:X-Forwarded-Proto} =http
RewriteRule .* https://%{HTTP:Host}%{REQUEST_URI} [L,R=permanent]
```

so that the HTTP-to-HTTPS-Redirect will only happen, if the request to the load balancer was HTTP.  
In this case your load balancer has to send the X-Forwarded-For and X-Forwarded-Proto HTTP header.

### LoginMaxAttempts

The number of failed login attempts of a particular user 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).

### 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.

## 14.1.9 TDNS Settings

All Registration Servers are part of a network of Registration Servers managed by a single TDNS (TeamDrive Name Server).

To add a Provider or Registration Server to the network, they must first be added to the TDNS directory. This is done on the Master Registration Server of the network, in the Admin Console under “Manager Servers”.

A “Checksum Key” is then generated by TDNS, which is required to create the Provider or to install the new Registration Server.

### TDNSURL

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

If this the URL is set to use HTTP, then it will be changed to HTTPS when upgrading to Registration Server version 4.5.6. This is a once-off change, however HTTPS should be used for security reasons, and HTTP access to TDNS will be deprecated and will be disallowed in the future.

### UsersPerSyncCall

The number of users that are sent per request when syncing the user list with TDNS.

## 14.2 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 83)) via the Administration Console, the new Provider’s settings can be changed by clicking **Providers -> Provider Settings**.

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

### 14.2.1 ADMINCONSOLE Settings

#### ADMIN\_CONSOLE\_SEND\_EMAIL

If set to `True` the Admin Console will send email notifications by default.

#### ADMIN\_LICENSE\_REFERENCE

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

#### LOGIN\_IP

A comma-separated list of IP addresses allowed to login to the Admin Console. If empty, there are no restrictions to login other than those specified by the `PROVIDER_LOGIN_IP` setting (see below).

Note that if you wish to allow normal users to access the Admin Console, for example account managers, then it may be required to set this setting to empty, since the IP addresses used by account managers to access the Admin Console may vary. In this case, you may wish to set the `PROVIDER_LOGIN_IP` setting in order to restrict the users that have Provider level privileges or higher.

#### LOGIN\_TWO\_FACTOR\_AUTH

Set to `True` to enable two-factor authentication (2FA) via email for logging into the Administration Console.

The 2FA required for the Admin Console is not related to personal 2FA that may be required at the Account or User level. If 2FA has been enabled for an individual user, then the additional Admin Console 2FA will not be required as of Registration Server version 4.7.

#### PROVIDER\_LOGIN\_IP

This setting is similar to `LOGIN_IP` but it only restricts the login of users with Provider or higher privileges levels. The IP address of these users must be in the comma-separated list of IP addresses specified by this setting in order to login to the Admin Console.

If the setting is empty, then login of users with Provider privilege is not restricted.

### 14.2.2 API Settings

#### 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\_NOTIFICATION\_URL

When user change notification is enabled (see [API\\_ENABLE\\_NOTIFICATIONS](#) (page 90)), 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 236).

## 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 130) 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 72).

Note that if `API_SEND_EMAIL` is set to `False`, then users created using the [registeruser](#) (page 140) API call will be automatically activated, if the `<activate>` tag is not explicitly set. This is to avoid having to send an activation email to the user.

## API\_ENABLE\_NOTIFICATIONS

Set this setting to `True` to enable user change notifications. When enabled you must also set [API\\_NOTIFICATION\\_URL](#) (page 90).

See [User Change Notifications](#) (page 236), for more details.

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

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

## 14.2.3 AUTHSERVICE Settings

These settings are used to configure access to an external Authentication Service (see [External Authentication](#) (page 11)).

### AUTH\_CHANGE\_EMAIL\_URL

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

### 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/yvva/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/yvva/portal/register.html`

## **AUTH\_SERVICE\_NAME**

This setting specifies the name of the External Authentication Service to be used by users of the Provider by default. See [Services](#) (page 60) for details on how to create a named authentication service.

AUTH\_SERVICE\_NAME is used when USE\_AUTH\_SERVICE is True or when external authentication has been explicitly enabled for a user.

The default External Authentication Service can also be overridden on User level, by specifying a specific External Authentication Service for the user. This is usually done automatically when the user uses an email address of a registered domain.

If AUTH\_SERVICE\_NAME is empty, and external authentication is enabled for the user then the Registration Portal Login pages are used. This is a URL of the form:

`http://<reg-server-host>/portal/login.html`

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

Set this value to the URL that references 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/yvva/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 prompted to re-enter their password.

Note that this setting applies to all users, not only to those using external authentication.

## **DEFAULT\_AUTH\_SERVICE\_NAME**

This setting contains the default name of an “unnamed” External Authentication Service. It must be used if the Provider is using an External Authentication Service that has not been upgraded and therefore does not return its service name.

An authentication service is assigned a name when it is registered under the “Manage Domains and Services” Provider page in the Admin Console. When this is done, certain email domains are also assigned to the External Authentication Service.

When a service is registered for an existing authentication service that has not been upgraded you must set the DEFAULT\_AUTH\_SERVICE\_NAME setting, or users will get an error on login.

## **PREVIOUSLY\_UNNAMED\_SERVICES**

This is a comma separated list of registered (named) external authentication services.

Add the name of services to the list when upgrading existing authentication services to a named External Authentication Service. Named services are registered using the Admin Console.

This list of services is used by the Registration Server to identify users that were registered using an External Authentication Service, before upgrade, and are therefore not associated with a named authentication service.

Note that users of such an upgraded service will not be able to login until the name of the service has been added to this list.

### ENABLE\_PORTAL\_PAGES

Set `ENABLE_PORTAL_PAGES` to `True` in order to enable the Registration Server Portal Login Pages. These pages provide a Web-based login for TeamDrive clients if required.

By default, `ENABLE_PORTAL_PAGES` is `True` if `USE_AUTH_SERVICE` is `True`.

Note: if `USE_AUTH_SERVICE` is `True` but the Portal Pages are not being used we recommend that you set `ENABLE_PORTAL_PAGES` to `False` explicitly.

### USE\_AUTH\_SERVICE

Set to `True` in order to set the default External Authentication Service. Note that if `USE_AUTH_SERVICE` is set to `False`, external authentication can still be used by users of the Provider that have been explicitly assigned an External Authentication Service or if external authentication has been specifically enabled for the user. This is usually by association with an particular email domain.

The default External Authentication Service is specified by the `AUTH_SERVICE_NAME` setting (see [AUTH\\_SERVICE\\_NAME](#) (page 91) for more details).

If `AUTH_SERVICE_NAME` is empty, and `USE_AUTH_SERVICE` is `True` then the Registration Portal Login Pages are used. This is a URL of the form:

`http://<reg-server-host>/portal/login.html`

In this case, `ENABLE_PORTAL_PAGES` must be set to `True`. `ENABLE_PORTAL_PAGES` is `True` by default if `USE_AUTH_SERVICE` is `True`.

When external authentication is enabled, the settings: `AUTH_CHANGE_EMAIL_URL`, `AUTH_REGISTER_URL` and `AUTH_SETUP_2FA_URL` are also active.

## 14.2.4 CLIENT Settings

### 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 115) 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.

This setting will not affect users that are using an External Authentication Service.

## 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 78).

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 78)) 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 112)

## 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 record to identify user 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 113)). 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 using 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 113)). Store forward invitations are only sent to globally visible email addresses.

### MAXIMUM\_DEVICES\_PER\_USER

This setting specifies the maximum number of user devices that may be activate at any given time. By default the value is zero which means there is no limit. This setting is new in Registration Server 4.5.0.

If set to another value the new “Deactivate/Activate Devices” auto task (see `deactivate_activate_devices_task` for more details) will enabled and disable devices as required to ensure that only the specified number of devices are active.

The disabled devices are set to the “too many devices” status, which means that the client user interface and synchronisation will be disabled. In addition, the device will not receive invitations, until it is reenabled.

The Registration Server always disables the least recently used devices. As a result, a device can be reenabled by simply starting the TeamDrive client. However, it takes an average of 3 hours before a device is reenabled by the server.

If the activation of devices is changed then the server sends an email to the user using the **devices-disabled** email template (see [Templates for Client Actions](#) (page 69)).

As of Registration Server version 5.0.1 you can indicates that the device limit is a “soft limit” by prefixing the value with a ‘~’ character. For example: “~5” means a soft limit of 5 devices per user. Soft limit in this case means that the limit is only enforced if a user does not already exceed the specified limit.

The purpose of this feature is to prevent significant disruption of the TeamDrive service for users that are currently using an excessive number of devices. Effectively the soft limit allows time to deal with violations of the terms of agreement on a case by case basis.

### MAXIMUM\_OUTLOOK\_ADD\_INS

This is the maximum number of Microsoft Outlook Add-ins that can be installed per user. The default value is 1. This is the minimum value that may be set.

If the user registers more Outlook Add-ins than specified by this value, then old registrations will be automatically deleted. This makes the old installations unusable.

Note that if the value of `MAXIMUM_OUTLOOK_ADD_INS` is reduced, the Registration Server will not reduce the number of Add-in registrations of users until a new Outlook Add-in is registered.

### 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.

### 14.2.5 CSVIMPORT Settings

Users can be created by importing a CSV file. The CSV file can either be 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 `admin_console_csv_user_imports`.

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 `csv_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.

## 14.2.6 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\_SENDER\_EMAIL

Email address of the 'envelope-from'-email in user-to-user mails like invitations and 'from'-email for all server-to-user emails like the activation email, new password, etc, if empty the MailSenderEmail global setting value will be used. The address will also be used to set the "sender header" (see *MailSenderEmail* (page 80)).

### FROM\_EMAIL\_OPTIONS

This setting determines the "From:" email address used when sending invitations to TeamDrive spaces, and other notifications sent directly from TeamDrive users.

The option can be set to one of the following:

replyto-only: "From:" is set to the EMAIL\_SENDER\_EMAIL setting value, and "Reply-To:" is set to the sender email address.

replyto: "From:" is set to the sender email address, followed by EMAIL\_SENDER\_EMAIL which is placed in angle brackets (< and >).

"Reply-To:" is then set to the sender email address.

For example: if td-user@example.com is the sender email address, and EMAIL\_SENDER\_EMAIL is set to no-reply@teamdrive.com, then "From:" is set as follows:

From: td-user@example.com <no-reply@teamdrive.com>

The purpose of this setting is to prevent problems with Mail Servers that generate an error when the "From:" email address is unknown. The use of angle brackets is to ensure that email programs display the actual email address of the sender, as some email programs do not display the "Reply-To:" address automatically.

replyto-via: This option is the same as replyto, but with the addition of the text: (via BRAND) to the "From:" header, for example:

From: td-user@example.com (via BRAND) <no-reply@teamdrive.com>

provider: "From:" is set to the EMAIL\_SENDER\_EMAIL setting value, and the sender email address is ignored.

user: "From:" is set to the sender email address, and "Reply-To:" is not set.

The default value for this setting is replyto-via.

## 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.

## SENDER\_HOST

Host name of the email originator. If empty the `MailSenderHost` global setting value will be used. Will be visible in the email header in 'Received: from'. If using an own Host Name, the IP address must match to the servers FQDN (see [MailSenderHost](#) (page 80)).

## SMTP\_SERVER

The SMTP Mail Server address (host name), if empty the `SMTPServer` global setting value will be used.

In order to use a TLS/SSL connection to the SMTP server prefix the host name of the server with “smpts” protocol, for example: “smpts://my.smtpserver.com”. If no protocol is specified then “smtp” is assumed.

## SMTP\_SERVER\_USER

An username for smtp authentication.

## SMTP\_SERVER\_PASSWORD

The password for smtp authentication.

## SMTP\_SERVER\_TIMEOUT

the Timeout in seconds when waiting for the SMTP Mail Server, if empty the `SMTPServerTimeOut` global setting value will be used.

## 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.

Note that support uploads will not be allowed if this setting is empty.

## 14.2.7 HOSTSERVER Settings

A TeamDrive Enterprise Host Server can be registered with a Registration Server and assigned to a particular Provider. This is done during the setup of the Host Server.

In the Admin Console, the Default Provider can view a list of Host Server available to them. The “Activation Code” in the Host Server list is required to complete the registration of a Host Server.

Once registered a Host Server can be selected for usage by users by default (see `HOST_SERVER_NAME` below). This is done by creating a **default depot** for all new users (see `HAS_DEFAULT_DEPOT` below).

Host Server can also be assigned to specific account for Account level usage. In this case, the Account level Host Server will be used in place of the Host Server specified by the `HOST_SERVER_NAME` setting.

### ALLOW\_SPACE\_NAME\_STORAGE

If set to `True` the owner or manager of a Depot will be able to enable Space name storage for the Depot. When enabled, the name of Spaces are stored on the Host Server. The Space list of the Depot will include the name.

The advantage of Space name storage is that the names of the Spaces is displayed in the Depot Space list in the Registration Server and Host Server Admin Console. The disadvantage is that unauthorised managers or administrators may have access to this information.

If you disable Space name storage, Space names will be deleted from the Host Server database 7 days later (requires Host Server version 5.1).

Note that the Space names are only transferred for active Spaces on active devices. This means that if you disable Space name storage, the names of inactive and unused Spaces will never be stored again. In addition, Space name transfer requires TeamDrive Client version 5.3 or later.

### 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`.

Since Registration Server version 4.6.0 this value is `True` by default.

### HAS\_DEFAULT\_DEPOT

Set to `True` if a default depot should be created for all new users. The `HOST_SERVER_NAME` setting specifies the Host Server to be used to create the default depot. If the user is an account member, and the account has an Account level Host Server, then this Host Server will be used instead.

Note that a default depot is only created if a user does not otherwise have a depot in use. A user may be assigned a depot automatically due to the following:

- The `PROVIDER_DEPOT` setting specifies a depot to be assigned to all new users of a Provider.
- The user belongs to an account where the manager has specified an Account level default depot.

It is also possible to manually assign a depot to a user for usage.

In addition, a default depot will not be created for a user due to the following:

- The user has a license with the `NoDepot` feature (see [DEFAULT\\_FREE\\_FEATURE](#) (page 102) for details).
- The `HAS_DEFAULT_DEPOT` is overriding on the Account level (see below).

If an account has its own Host Server, then the manager can also override this setting by setting the default depot handling at the Account level. The options are:

- Use the Provider level defaults
- Never create or assign a default depot
- Always create a default depot if the user has no depot

### HOST\_DEPOT\_SIZE

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

If an account has its own Host Server, then the manager can also override this setting and set the storage size of default depots at the Account level.

## HOST\_SERVER\_NAME

Specifies the Host Server to be used when creating a default depot for new users.

If the user is a member of an account, and the account has an account level Host Server, then this value is ignored.

## HOST\_TRAFFIC\_SIZE

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

If an account has its own Host Server, then the manager can also override this setting and set the traffic limit of default depots at the Account level.

## PROVIDER\_DEPOT

This setting is used to specify that a specific depot is to be assigned for usage by all new users of the Provider.

The depot is assigned to the user in place of, for example, creating a default 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. This is done automatically when using the Admin Console.

Setting `PROVIDER_DEPOT` to zero does not remove the depot from users already using the specified depot, it just prevents the depot from being assigned to user's in the future.

At the Account level it is possible to override this setting by setting an Account level default depot. The options at the Account level are:

- Use the Provider level defaults
- Never create or assign a default depot
- Always create a default depot if the user has no depot

In addition, if the user's license has the `NoDepot` feature, then this setting will also be ignored (see [DEFAULT\\_FREE\\_FEATURE](#) (page 102) for details).

## 14.2.8 INVITATION Settings

### ACTIVATE\_ON\_INVITATION

The setting determines whether a user account is activated after setting their password after receiving an **inv-newuser-invited** email (see `INVITATION_CREATES_USER` below).

The default value is `True`, which means that the user account will be activated when the user sets his password. If set to `False`, then the user is required to activate their account when they login for the first time.

### AUTO\_CREATED\_USER\_TIMEOUT

This setting specifies a certain number of days. If an automatically created user is not activated within the time specified here, then the user is automatically deleted.

Note that the user is only deleted if the user account is not modified in any way, and the user does not login to TeamDrive using the user account.

The default value is 60 days. Setting `AUTO_CREATED_USER_TIMEOUT` to zero disables the deletion of users.

### FORWARD\_INVITATION\_TIMEOUT

This setting specifies the time (in minutes) that a forwarded Space invitation is retained, after the user that is to receive the invitation has registered. The default is 14 days. This ensures that a user will receive the invitation after registration, even after the first installation is unsuccessful for some reason.

To be more specific, when an unknown user is invited to a Space there it is not possible to send the invitation to the user directly. Instead, the invitation is stored as a “store-forward” invitation until the user registers. The time a store-forward invitation is retained is specified by the global `InvitationStoragePeriod` setting. If the user registers before this time is expired then the invitation is forwarded to the user’s TeamDrive Client installation.

Previously, the store-forward invitation would then be deleted. However, this can lead to a problem, if the invitation is not or cannot be immediately accepted. In this case the invitation may be lost.

In Registration Server version 5.0.2, the store-forward invitations are now retained after the first installation of the user. The retention period is determined by the `FORWARD_INVITATION_TIMEOUT`. Any new installations in this time will receive the invitation, no matter whether the invitation was accepted on a previous installation or not.

### INVITATION\_CREATES\_USER

In the TeamDrive client it is possible to invite a user (via email) to join a space, even when the user is not yet a registered TeamDrive user.

When this setting is set to `True` the Registration Server automatically registers these users using the email address used in the invitation.

An email using the **inv-newuser-invited** is sent to the new user with the details of the space to which the user is invited, and a link which can be used to activate the new account (see *Templates for Client Actions* (page 69)).

This has the advantage that the user can be on-boarded quickly, and need only set a password in order to activate their account. After this, the user can be directed to an online web portal, or to a download page for TeamDrive client.

If `ACTIVATE_ON_INVITATION` is `False` (see above), then the user will be required to activate their account after the first login.

### INVITATION\_NEW\_USER\_PROVIDER

When `INVITATION_CREATES_USER` is enabled, this setting determines the Provider with which the user is registered. By default the new user is registered with the same Provider as the inviting user.

### ISOLATED\_INVITATION\_ZONE

This setting specifies a group of Providers as a comma separated list of Provider codes (REGSERVER-1771). When set, users of the Provider cannot invite anyone that is not a user of one of the Providers in the group.

In addition users of this Provider cannot be invited by users of Providers not in the group.

For example: assume user A is a user of Provider AAAA, and user B is a user of Provider BBBB, and C of CCCC.

If `ISOLATED_INVITATION_ZONE` of Provider AAAA is set to AAAA,BBBB, and `ISOLATED_INVITATION_ZONE` of Provider BBBB is also set to AAAA,BBBB then: user A can invite B but cannot invite C and user B can invite A but not C. In addition, user C cannot invite user A or B.

### MAX\_PROMOTION\_USER

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

You can configure a referral program by setting this value to a value greater than zero.

A referral program provides an incentive for users to invite other users in order to increase their free storage limit (see `PROMOTION_UPGRADE` below).

---

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

- The invited user is not registered before being 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 is activated, increasing the depot values and sending the notification mails to the inviter (see *Templates for Client Actions* (page 69)).

This feature requires an active Host Server and default depots for your users (see above *HOSTSERVER Settings* (page 97)).

## NEW\_USER\_LICENSE\_FEATURES

When `INVITATION_CREATES_USER` is enabled, this setting determines the features of the license created for the new user.

By default this is set to: **Professional, Restricted, NoDepot**, which means the user has professional account which is restricted to accessing a limited number of spaces (see *active-spaces-limit (default: 0)* (page 113)), and no default depot is created for the user.

For a list of options available see *DEFAULT\_FREE\_FEATURE* (page 102).

## PROMOTION\_UPGRADE

If you are using a referral program then this is the upgrade size in bytes that of a user’s default depot for each user invited.

The depot limit is increased for both users: the inviter and invitee.

## 14.2.9 LICENSE Settings

### ACCOUNT\_RESTRICTIONS

This setting specifies license based restrictions to user accounts.

Current the only setting supported is `super-pin-repo-pro-license-limit`, which is used to restrict the use of the Super PIN Repository to accounts with a certain number of professional licenses. This is all license with the `professional`, `secureoffice` and `agent` license features.

For example, setting this setting to `super-pin-repo-pro-license-limit=5` will disable the Super PIN Repository unless the account has 5 or more professional licenses. By default, the use of the Super PIN Repository is not restricted.

### ACTIVE\_SPACES\_LIMIT

This setting specifies the maximum number of Spaces that can be active in the TeamDrive Client for users that have a license with the **Restricted** feature.

By default the value is set to 1. This limitation can be disabled by setting the value to 0.

If non-zero the Registration Server will automatically add the `active-spaces-limit` setting to the `CLIENT/CLIENT_SETTINGS` value sent to the client, provided the `CLIENT_SETTINGS'` value does not explicitly include this setting.

### ALLOW\_CREATE\_GROUP

This setting determines whether groups can be created by users or managers of the Provider (REGSERVER-1735). By default the value is `False`.

This setting can only be modified by users with SUPER-USER privileges, or a manager of the default Provider (see *DefaultProvider* (page 83)).

If the `ALLOW_CREATE_GROUP` is `False`, then creation of groups can be enabled explicitly on an account by a superuser. In this case, account managers can create groups for that account.

### ALLOW\_CREATE\_LICENSE

Set to `True` to allow the creation of licenses for this Provider. This setting can only be changed by a user with SUPER-USER privileges, or a manager of the default Provider (see *DefaultProvider* (page 83)).

### ALLOW\_MANAGE\_LICENSE

Set to `True` to allow the management of licenses for this Provider. This setting can only be changed by a user with SUPER-USER privileges, or a manager of the default Provider (see *DefaultProvider* (page 83)).

### DEFAULT\_ACCOUNT\_FEATURE

The `DEFAULT_ACCOUNT_FEATURE` determines the features of default license of users that belong to an account. This setting is similar to the `DEFAULT_FREE_FEATURE` setting which applies to users that do not belong to an account.

`DEFAULT_ACCOUNT_FEATURE` is set to **Personal** and **Restricted** by default. These and other details about license features are described in the section: *DEFAULT\_FREE\_FEATURE* (page 102) below.

If `DEFAULT_ACCOUNT_FEATURE` is empty then the Admin Console will not allow managers to create a new license when adding a user.

### DEFAULT\_FREE\_FEATURE

This setting determines the features of the default license of users that do not belong to an account (see *DEFAULT\_ACCOUNT\_FEATURE* (page 102) for the setting for account users). It is set to **WebDAV** by default.

When a user is created or registered for the first time, and no license is specified, a default license is automatically created for the user. The settings `DEFAULT_FREE_FEATURE` and `DEFAULT_ACCOUNT_FEATURE` determine the features of this license, depending on whether the user is a member of an account or not.

Note that if the setting `DEFAULT_LICENSEKEY` is set, then a default license will never be created (see *DEFAULT\_LICENSEKEY* (page 104) below).

Due to license changes between TeamDrive 3 and TeamDrive 4 there are differences in the meaning of the license features between these versions.

TeamDrive 3 supports two commercial license models: the Personal and the Professional Licenses (these are identified by the **Personal** and **Professional** license features).

Licenses without these features are considered free licenses by TeamDrive 3, which then imposes a blanket limit on the amount of data handled by the client (set to 2 GB by default). The TeamDrive 3 Personal and Professional Licenses remove this restriction.

However, the TeamDrive 3 Personal License disables certain features only available to the Professional License holder, this includes: a limit to the number of versions stored on the Host Server, publish file functionality is disabled and various email notifications and support for network drives is disabled.

TeamDrive 4 distinguishes between commercial/business and non-commercial users. TeamDrive 4 is free for non-commercial usage. Commercial and business users must purchase a Professional License.

TeamDrive 4 requires that non-commercial users confirm their non-commercial status daily, but otherwise imposes no restrictions on the non-commercial users.

Free commercial licenses are also available for TeamDrive 4, but these licenses must include the **Restricted** feature (see below).

Feature descriptions:

### **Agent**

The Agent feature is required by licenses used by the TeamDrive Agent.

### **Banner**

The Banner feature is which was only supported by TeamDrive 3 clients is no longer supported by Registration Server version 4.1 or later.

### **Inbox**

The Inbox feature is required by licenses used for a user hosting an Inbox.

### **NoDepot**

This license feature disables the automatic creation and assignment of a depots for a user. This means that this feature overrides the `PROVIDER_DEPOT` and `HAS_DEFAULT_DEPOT` Provider settings, and the Account level setting for creating a default depot.

This means that a new user with this license feature will only have a depot if the user is a member of an account with an Account level default depot.

### **Personal**

The Personal feature was used to create TeamDrive 3 Personal Licenses. Licenses for TeamDrive 4 clients should use the **Professional** feature instead. This feature bit is no longer supported by the Registration Server version 4.1 or later.

### **Professional**

The Professional feature is used to create TeamDrive Professional Licenses.

TeamDrive 3 Clients enabled certain Professional-only features when this feature is set.

TeamDrive 4 Clients disable the daily dialog which requires the user to confirm that he/she is non-commercial user of TeamDrive when this feature is set.

### **Restricted**

This feature enables restrictions that are specified using certain client settings. Currently the only active restriction is determined by the `ACTIVE_SPACES_LIMIT` Provider setting. This setting deter-

mines the maximum number of spaces that may be active on the client (see [ACTIVE\\_SPACES\\_LIMIT](#) (page 101)).

### SecureOffice

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

### 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.

### DEFAULT\_LICENSEKEY

Define a specific license that will be assigned to all users upon registration. This license's features will override the features defined in the `DEFAULT_FREE_FEATURE` and `DEFAULT_ACCOUNT_FEATURE` settings.

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.

This setting is `True` by default.

However, if you are upgrading from Registration Server 3.0.017 or earlier, this setting will be set automatically be set to `False` for providers that already have licences with expiry dates. This is because expiry was not implemented by this version of the server, so the setting is disabled in order not to disrupt potential users of such licenses.

### EXT\_LICENCE\_REF\_UNIQUE

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

If you set `ADMIN_LICENSE_REFERENCE`, 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.

### SPACE\_SIZE\_LIMIT

The maximum size in bytes of active spaces for users with restricted license and non-professional licenses. This limit refers to the size of the space on the Hosting Service. User will not be able to enter a space that exceeds this limit if they do not have the required license. In addition, spaces that exceed this limit are disabled in the client.

The default value is "0", which means that restriction is disabled.

## 14.2.10 LOGIN 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.

### 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 providers on this Registration Server are allowed.

This setting caters for providers that have a specific version of the TeamDrive Client and want to ensure that only this type of client is used by the providers'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.

---

### ALLOWED\_LOGIN\_ATTEMPTS

This setting determines the number of times a user may fail to login before the failed login timer is activated (see [FAILED\\_LOGIN\\_TIMER](#) (page 106)). The default value is 3.

Note that this also includes login attempts when logging in with a temporary password, after a user has lost their password.

When the number of allowed failed logins is exceeded, the server sends an email using the **too-many-failed-logins** email template to the user (see [Templates for Client Actions](#) (page 69)).

### 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 a user can register new users 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.

As of Registration Server version 4.5 the server will add `enable-registration=false` to the `LOGIN/PRE_LOGIN_SETTINGS` sent to the TeamDrive client. The client will then disable the registration dialog accordingly (see chapter *enable-registration=true/false/default (default: true)* (page 116) for details).

### FAILED\_LOGIN\_TIMER

This is the time in seconds the Registration Server will wait before allowing another login attempt, after the user has failed to login the number of times specified by *ALLOWED\_LOGIN\_ATTEMPTS* (page 105). The default value is 300 seconds (5 minutes).

### LOGIN\_WITHOUT\_ACTIVATION

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.

---

If you don't want to allow users to activate new devices themselves, set `MANUAL_ACTIVATION_REQUIRED` to `True` (see *MANUAL\_ACTIVATION\_REQUIRED* (page 106) below).

### MANUAL\_ACTIVATION\_REQUIRED

If you require manual activation of all TeamDrive devices then set this setting to `True`. In this case, whenever a user installs a new device they will receive an email indicating that manual activation is pending.

In addition, an email notification is sent to the users specified by the `NEW_DEVICE_NOTIFICATION_LIST` Provider setting (see below). The email includes the details of the new device and link which directs the manager to the device on the Admin Console.

The user will not be able to proceed to use the new installation until the device has been manually activated on the Admin Console.

---

**Note:** In order to enable manual activation, the Provider settings: `LOGIN_WITHOUT_ACTIVATION`, `SUPERPIN_LOGIN_WITHOUT_ACTIVATION` and `ACTIVATE_ON_INVITATION` must be set to `False`.

---

See *requiring\_manual\_activation* for details.

### NEW\_DEVICE\_NOTIFICATION\_LIST

This is a comma separated list of usernames and/or email addresses of existing users that must be notified when a new device is created.

This list can also include the username of a Provider manager which is specified in the "Provider Record" on the Provider Setting page in the Admin Console. Alternatively you can specify the Provider administrator by using

the Provider Code preceded by a “\$” characters. For example if you want to notify the Administrator of Provider ABCD, then you can add “\$ABCD” to the list.

If `MANUAL_ACTIVATION_REQUIRED` is set to `True` and this setting is empty, a notification will be sent to the Provider administrator.

With regard to using this setting to perform manual activation of devices see `manual_activation_users` for further details.

## **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.

## **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 118).

## **SUPERPIN\_LOGIN\_WITHOUT\_ACTIVATION**

On login to a new installation with the Super PIN instead of a password, this setting determines whether an activation email is sent to the user or not.

By default the value is `False`, which means that activation of the new installation is required, and an email with an activation link will be sent.

## **TEMP\_PASSWORD\_CHARACTERS**

A string of characters that includes all characters used to generate a temporary password. By default this includes digits and letters excluding characters that are “ambiguous” because they may be mistaken for another character, including: 0, l, z, I, L and O. This is done to make the password easy to copy manually if necessary.

## **TEMP\_PASSWORD\_LENGTH**

The length of the temporary password used to set a new password. The default is 6 characters as of Registration Server version 4.7.

## **TEMP\_PASSWORD\_TIMEOUT**

This is the time in seconds that a temporary password is valid, The default value is 10 minutes. The minimum value is 1 minute, and the maximum is 2 hours.

A temporary password is sent to the user via email when setting a new password during login. If the temporary password is not used within the timeout specified here, it is marked as invalid, and the user must request a new temporary password.

Note if the Super PIN functionality is enabled for a user account, then the password can no longer be set using a temporary password. Instead, the user must use either their Super PIN or a Recovery Code obtained using the Recovery URL (a QR code) in order to login and change their password.

## USER\_IDENTIFICATION\_METHOD

This setting determines how a user is identified. 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.

As of Registration Server version 4.5 the server will add `user-ident-method=<value>` to the `LOGIN/PRE_LOGIN_SETTINGS` sent to the TeamDrive client. The client will then adjust the registration dialog accordingly.

TeamDrive clients older than 4.6.9 do not support this option and will continue to require a username to be specified on registration, no matter what the value of this setting.

**username** This means that users 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 users are identified using the user's email address. In this case, the user 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. 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 a user, 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 for user identification 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 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 records, further identified by the Provider code.

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

### 14.2.11 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 Settings](#) (page 85)).

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\_FUSE**

This URL redirects to the Provider's fuse information 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. This setting is deprecated.

## **REDIRECT\_PURCHASE**

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

## **REDIRECT\_SECURITY**

This URL redirects to the Provider's information page how to join spaces that have certain security requirements.

## REDIRECT\_SUPERPIN

This URL redirects to the Provider's information page that explains the usage and consequences of activating the Super PIN.

## REDIRECT\_TERMS

This URL redirects to the Provider's "Terms of Service" page.

## REDIRECT\_TUTORIALS

This URL redirects to the Provider's tutorials page.

## REDIRECT\_USERINVITEUSER

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

## 14.2.12 SHOP Settings

If the Provider has an associated shop, then set the `SHOP_SERVICE_NAME` below appropriately. And set `SHOP_ENABLED` to `True`.

This will enable the display of buttons in the Admin Console that reference Licenses and depots in the shop. The license/depot must have an external reference, or the buttons will not be displayed.

The button will take the user to the associated page in the shop and automatically perform a login for the user.

### ENABLE\_PURCHASE\_LICENSE

Set to `False` to disable the purchase license button in the Admin Console. The default value is `True`.

### ENABLE\_PURCHASE\_DEPOT

Set to `False` to disable the purchase depot button in the Admin Console. The default value is `True`.

### ENABLE\_UPGRADE\_LICENSE

Set to `False` to disable the upgrade button for licenses in the Admin Console. The default value is `True`.

### ENABLE\_UPGRADE\_DEPOT

Set to `False` to disable the upgrade button for depots in the Admin Console. The default value is `True`.

### SHOP\_ENABLED

Set to `True` to enable shop buttons in the Admin Console. The default value is `False`.

### SHOP\_SERVICE\_NAME

This is the name of the Shop Service that is used by the Admin Console to initiate a shop-based function.

How to create a service is described in [Services](#) (page 60).

### 14.2.13 UPDATE Settings

The Registration Server informs the TeamDrive client if a software update is available. Use the following settings to control the update notifications. For more details on client updates see: `managing_client_updates`.

#### CURRENT\_CLIENT\_VERSION

This setting determines whether TeamDrive client will post an update notification or not.

If the version specified here is greater than the current TeamDrive client version, and the version is greater than the last version that the user choose to ignore, then an update notification will be displayed.

#### ENABLE\_UPDATE\_TEST

Set this value to `True` in order to test how the TeamDrive client reacts to an update notification.

You can test the update notification without triggering an update notification to all clients by setting this value to `True`, and by specifying a test user (`UPDATE_TEST_USER``) and a test version (``UPDATE_TEST_VERSION`).

#### UPDATE\_TEST\_USER

Use this setting to specify a user (either username or email address) for testing the update notification.

The specified user will then receive an update notification regardless of whether the update is required by their TeamDrive client or not (see `UPDATE_TEST_VERSION` for more details).

#### UPDATE\_TEST\_VERSION

Set this value to specify the test update version. If not set, the server will return the `CURRENT_CLIENT_VERSION` value.

If version returned is higher than the current version of the client then the client will always display the update notification.

If the not, then the client will only display the update notification if the version is higher than the last version the user choose to ignore.

### 14.2.14 WEBPORTAL Settings

Manage web portals associated with the Provider here.

#### 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 a web portal.
- `deny`: Web portal access is denied to all users.
- `permit-by-default`: Users are allowed to access Web Protals but this setting can be overridden at the User level. This is the dafault value for this setting. Note that before Registration Server 4.5.6 the default was `permit`.
- `deny-by-default`: Users are not allowed to access web portals, but this value can be overridden at the User level. Note that before Registration Server 4.5.6 this setting was called: `peruser`.

Modifying the User level web portal access can be done on the Admin Console on the edit user page. If `ALLOW_WEB_PORTAL_ACCESS` is set to `permit` or `deny` then access cannot be changed at the User level.

In addition, an account manager can disable web portal access for all account members, in the Admin Console, as long as `ALLOW_WEB_PORTAL_ACCESS` is not set to `permit`.

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 113) and on the web access rights for a space created with a client 4.3.2 or newer.

---

### 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. Providers 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.

### ENABLE\_INBOX\_SERVICE

Set to `True` if a web portal should be used to create inboxes for accounts. The setting `WEBPORTAL_API_URL` specifies the web portal to be used when this is enabled.

Note that the setting `WEBPORTAL_API_CHECKSUM_SALT` must also be set correctly for the inbox service to work.

### WEBPORTAL\_API\_CHECKSUM\_SALT

This is the value of the web portal `APIChecksumSalt` setting, which is used to access the web portal API.

### WEBPORTAL\_API\_URL

This is the URL of the main web portal used by the Provider. This web portal is used to make API calls in order to create an inbox. For this purpose, the `WEBPORTAL_API_CHECKSUM_SALT` must be set.

In addition, the URL is returned on request for the web portal redirect URL. In other words, the URL in this setting is returned when the web portal URL is requested for a Provider.

The web portal redirect URL is required if users login to the incorrect web portal. The web portal can detect this situation, after performing a TDNS lookup, and then will attempt to redirect the user to the correct web portal.

## 14.3 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 93) 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 107) for more details.

The following TeamDrive Client settings can be adjusted:

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

Limit the amount of active spaces in the client. This setting only has an effect if the “**Restricted**” license feature is set on the user’s license as described in registration server how tos/restricted license. 0 means unlimited.

This setting should not be set explicitly in `CLIENT/CLIENT_SETTINGS`. The `LICENSE/ACTIVE_SPACES_LIMIT` Provider setting automatically sets the setting when set to a non-zero value (see [ACTIVE\\_SPACES\\_LIMIT](#) (page 101)).

### 14.3.2 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 118), you should allow email login. Note that, although this is usually the case, the email field is not necessarily unique in TeamDrive. If the same email address is used by different users, the client will show a drop-down list of all possible users after the email address was entered.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 14.3.3 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`.

### 14.3.4 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 116)). 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`.

### 14.3.5 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`.

### 14.3.6 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, virtual.

This setting may be used in `CLIENT_SETTINGS`.

### 14.3.7 auto-invite-users=list

A list of user names to be automatically invited into newly created Spaces with specified default-invitation-rights. 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.

### 14.3.8 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.

### 14.3.9 default-join-mode (default: default)

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

This setting may be used in CLIENT\_SETTINGS.

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

How man days will an unencrypted published file (see *enable-publish=true/false/default (default: true)* (page 116)) be kept on the server until it will autoamtically removed on the server. 0 means unlimited and the user has to unpublish the file by himself.

This setting may be used in CLIENT\_SETTINGS.

### 14.3.11 default-server-mode (default: default)

Defines if the default server for creating spaces can be modified by the user or not. The following values are allowed:

- **default** (default): means that the users selected server is used, unless he selects the same as the server in which case the user selection is cancelled
- **ignore-server**: means that the user selected default server is never changed by the server
- **ignore-user**: means that the server default server is used, and the user cannot change the selected server
- **use-selected**: means that the server, per user, selected server is used, and the user cannot change the selected server

This setting may be used in CLIENT\_SETTINGS.

### 14.3.12 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.

### 14.3.13 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 118). 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`.

### 14.3.14 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`.

### 14.3.15 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`.

### 14.3.16 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`.

### 14.3.17 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`.

### 14.3.18 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`.

### 14.3.19 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`.

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

Enable/disable the Key Repository. If enabled, the user's space keys are stored encrypted using the user's password on the server. When the user installs TeamDrive on a new device, the space keys are retrieved from the Key Repository and the user is able to activate the spaces in the new installation.

If the Key Repository is disabled, after installing a new device, the user must be re-invited to all spaces in order to gain access to the spaces on the new device. If the user already has a TeamDrive installation he/she can do a self-invitation.

This setting may be used in `CLIENT_SETTINGS`.

### 14.3.21 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`.

### 14.3.22 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`.

### 14.3.23 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`.

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

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

If disabled, users must be created using the Registration Server API or a user import script as described in `importing_users_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 11)).

This setting may be used in `PRE_LOGIN_SETTINGS`.

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

Enables/disables setting the license key in the client.

This setting may be used in `CLIENT_SETTINGS`.

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

Defines the default value for newly 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` can't 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`.

### 14.3.27 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`.

### 14.3.28 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`.

### 14.3.29 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`.

### 14.3.30 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`.

### 14.3.31 inbox-url=URL

The URL for the inbox agent.

This setting may be used in `CLIENT_SETTINGS`.

### 14.3.32 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.

---

**Note:** In case of using the email as username (see [USER\\_IDENTIFICATION\\_METHOD](#) (page 108)) you have to use the magic username as inbox username.

---

This setting may be used in `CLIENT_SETTINGS`.

### 14.3.33 master-user=username

A single unique user name that will automatically be invited into every newly created Space with the MasterUser-Rights privilege (to automatically invite more than one user, use *auto-invite-users=list* (page 114)). The user must already exist with at least one activated device.

---

**Note:** In case of using the email as username (see *USER\_IDENTIFICATION\_METHOD* (page 108)) you have to use the magic username as master username.

---

This setting may be used in `CLIENT_SETTINGS`.

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

The Registration Server supports UTF-8 characters for usernames. If you upload users 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 107)).

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

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 14.3.35 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`.

### 14.3.36 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`.

### 14.3.37 spaces-path

Default path for newly created Spaces by the user.

This setting may be used in `CLIENT_SETTINGS`.

#### **14.3.38 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

### 15.1 API Basics

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

- Obtain information about various objects and parameters managed by the Registration Server, including: Accounts, Users, Licenses and Depots.
- Make change and automate processes that would otherwise require manual changes using the Admin Console.

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 are accepted.

Only registered services can access the Registration Server API (*Services* (page 60)). All services have an “Authorisation Method” and an “IP Address List” that are relevant to API access. See *API Usage* (page 121) below for details.

#### 15.1.1 API Usage

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

```
https://<reg-server-host>/yvva/api/api.xml?checksum=<auth-hash>
```

Where:

- <reg-server-host> is the host name of the Registration Server you are connecting to.
- <auth-hash> is the “authorisation hash” value as described below.

#### Request Authorisation

That a service has the right to make an API request is verified using an “authorisation hash”. This value is calculated by hashing the request body (the POST content) and a key specified by the Registration Server.

The hashing algorithm used depends on the Authorisation Method set for the server. This may be one of the following: **MD5 (APIChecksumSalt)**, **MD5 (Endpoint specific key)** or **HMAC-SHA1 (Endpoint specific key)**, as described here: *API Access* (page 61).

The key can be obtained by clicking on the “Show Key” button of the service in the Admin Console. You need Provider level privileges to view the key.

For MD5-based authorisation methods, the hash value is calculated by concatenating the request body with the key and then calculating the MD5 hash value, and finally converting to lower case.

For example in PHP this would look as follows:

```
$auth_hash = md5($request_body.$service_key);
```

(Note: '.' is used for string concatenation in PHP).

For HMAC-SHA1 based methods the has value is calculated in PHP is as follows:

```
$auth_hash = hash_hmac('sha1', $request_body, $service_key);
```

In all cases the result is a lower-case hex string. The MD5 result is 32 characters long and HMAC-SHA1 result is 40 characters long. Th Registration Server requires a lower-case hex value. In the case of PHP it is not necessary to use the strtolower() function because the results of these functions is already a lower-case string.

As described above the authorisation hash value must be added to the URL of the API request as a “search arg”. In PHP, for example, as follows:

```
$api_url = 'https://'.$reg_server_host.'/yvva/api/api.xml'  
$api_url = $api_url.'?checksum='.$auth_hash;
```

### IP Address List

A service has a number of IP addresses, which are used to verify the origin of a API request.

An IP address must uniquely identify a service, so it is not possible to run multiple services on the same host machine.

A service that accesses the API with an unknown IP address will receive an “Access Denied” error.

### Admin Console API Usage

The Admin Console also accesses the Registration Server API in order to perform a number of functions. For this reason the AdminConsoleIPAddress global setting must be set to the IP address of the Admin Console.

This will normally be done during Registration Server setup. If, for some reason, this value was not set correctly you will not be able to login to the Admin Console. In this case see admin-console-no-access for assistance.

Note that is want to be to manager multiple Providers in the Admin Console then you must set APIAllowSettingDistributor to True as described below.

### API Access Rights

API access rights are at the level of the Provider. That is, the API user is authorised to perform all action that can be performed by an administrator with Provider level privileges. For this reason, only fully trusted sources should be allowed to access the Admin Console.

When a service accesses the API it is authorised as the Provider to which the service belongs. The authorised Provider can be changed by setting an alternative Provider code in the <distributor> tag.

But this is only permitted when the following applies:

- APIAllowSettingDistributor is set to True and,
- the authorised Provider is the so-called “Default Provider”, which is the Provider that has authority over the entire Registration Server.
- or alternatively, the Provider in the <distributor> tag is “managed by” authorised Provider.

The “managed by” relationship can be set in the Admin Console. This feature allows a group a Providers to be managed by a single “manager Provider”. See “Providers” -> “Provider Settings”.

## Usage Considerations

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.

If your service provides access to endusers, for example, a Shop Service, then it is the responsibility of your Application Server to ensure that user's are only able to access their own data.

As mentioned above the authorisation of the API is at the Provider level, and therefore the API does not enforce User level access restrictions.

### 15.1.2 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 77)).

#### 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 enduser (see the [registeruser](#) (page 140) 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 93)) 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).

**<activationcode>:** **This option is new in Registration Server 4.0. It can only** be used to identify a user if the user has not yet been activated. If the user has already been activated a **-30100** error will be returned.

Note that the activation code may only be used to identify the user in API calls that explicitly allow this (check the call description for details).

#### 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 104)) to `True`.

### 15.1.3 The <origin> tag

The `<origin>` is new in the Registration Server version 4.0. The tag indicates the origin of the API call. By default this is “API-CALL”, but may also be: “LOGIN-PORTAL”, “ADMIN-CONSOLE”, “CLIENT-CALL” or “USER-IMPORT”.

Note that it is not recommended that you set this value unless you wish to simulate a different origin for debugging purposes.

The origin of a call may determine whether an email is sent by the API call or not. If the origin is not “API-CALL”, then by default an email will be sent (see [The <sendmail> tag](#) (page 124) for details).

This tag also determines which the type of email templates used for emails sent by the API. If the origin is “API-CALL” then “web-” type email templates are used otherwise, the “reg-” type emails are used.

### 15.1.4 The <sendmail> tag

The `<sendmail>` tag indicates whether an email may be sent by the API call or not. A number of API calls generate emails, and this email can be suppressed by setting `<sendmail>` to `false`.

If `<sendmail>` is omitted the default behaviour depends on the call. Check the description of the API call which specifies the default if the tag is omitted.

For those calls that do not have an explicit default, the Provider setting `API/API_SEND_EMAIL` ([API\\_SEND\\_EMAIL](#) (page 90)) determines if the call will send an email or not.

However, this setting is only used if the call origin (see [The <origin> tag](#) (page 124) above), is “API-CALL” (which is the default).

### 15.1.5 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/yvva/api/api.xml"
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"
```

### 15.1.6 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:

`https://<domain>/yvva/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/2.4.48 (CentOS) Server Port 443</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 235)).

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>
```

## 15.2 Registration Server API Calls

The following is list of all API calls.

### 15.2.1 getsettings

Use this call to retrieve Registration Server and Provider settings.

This call is new in Registration Server 4.5.1.

The `<settings>` tag must contain a comma separated list of settings to retrieve. Currently only `RegServerName`, `ClientSettings`, `CLIENT_SETTINGS`, and `PRE_LOGIN_SETTINGS`. Further settings will be added as required.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>loginuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <settings>ClientSettings,CLIENT_SETTINGS,..</settings>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <settings>
    <ClientSettings>...</ClientSettings>
    <CLIENT_SETTINGS>...</CLIENT_SETTINGS>
    ...
  </settings>
</teamdrive>
```

A `<settings>` block is returned in the reply, containing one tag for each setting requested, containing the associated value of the setting.

### Error Cases

Errors returned by this call include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30144**: Setting does not exists or, access to setting not permitted

### 15.2.2 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>` (see *Identifying Users* (page 123) for details).

The `<tmppassword>` tag is new in Registration Server 4.0. Setting this field allows you to attempt a login using a temporary password issued by the Registration Server after a lost password request (see the *sendpassword* (page 148) API call). In this case, `<password>` specifies the new password of the user.

In addition, the `<sendmail>` tag indicates whether the user receives a **passwd-changed** email or not. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

See *The <origin> tag* (page 124) for details on the `<origin>` tag.

`<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.

The tag `<includegroup>` (version 4.0) is optional. The default value is `true`. If the value is `true` the `<group>` tag will be included, if the user is a member of a group.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>loginuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <tmppassword></tmppassword>
  <licensereference></licensereference>
  <includegroup>true|false</includegroup>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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>
    <clientsettings></clientsettings>
    <keyrepository>true|false</keyrepository>
    <newsletter>true|false</newsletter>
    <emailbounced>true|false</emailbounced>
    <webportal>true|false</webportal>
    <group>
      <distributor></distributor>
      <groupname></groupname>
      <groupreference></groupreference>
      <manager></manager>
      <manageremail></manageremail>
      <groupcreated></groupcreated>
      <groupmodified></groupmodified>
      <groupdepot>
        <depotname></depotname>
        <depotreference></depotreference>
        <hosturl></hosturl>
        <depotid></depotid>
        <globalid></globalid>
        <username></username>
        <accountkey></accountkey>
        <accountreference></accountreference>
        <contractnumber></contractnumber>
        <storagelimit></storagelimit>
        <transferlimit></transferlimit>
        <created></created>
```

```

        <etl></etl>
        <status></status>
        <storageused></storageused>
        <transferused></transferused>
    </groupdepot>
    <licensekey></licensekey>
    <licensereference></licensereference>
    <clientsettings></clientsettings>
</group>
</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 enduser (see [registeruser](#) (page 140) 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`.
- `<clientsettings>` (version 4.0): User-level clients settings ( see [registeruser](#) (page 140)).
- `<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.
- `<group>` (version 4.0): This tag is included if `<includegroup>` is `true`, and the user is a member of a group. The `<group>` tag fields are described in the [getgroupdata](#) (page 232) call. A `<memberlist>` block is only included in the `<group>` tag returned by the “getgroupdata” call.

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 130)
- **-30101**: Wrong password
- **-30120**: User has been deleted

- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30137**: Too many failed login attempts

### Redirect due to user belonging to another Provider

If the Provider setting `API/API_REDIRECT` (see [API\\_REDIRECT](#) (page 90)) 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-direct 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>
```

## 15.2.3 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.

The `<email>` tag can be used to search for an email only. Alternatively you can use the `<useroremail>` tag, and set `<lookupboth>false</lookupboth>`. In this case the server will check if the lookup value contains a “@” character. If so, an email lookup will be done, otherwise a username lookup. This functionality was added in Registration Server 4.5.4.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>tdnslookup</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <email></email>
  <useroremail></useroremail>
  <lookupboth>true|false</lookupboth>
  <reference></reference>
  <authid></authid>
  <password></password>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <regserver>
    <distributor></distributor>
    <servername></servername>
    <domain></domain>
  </regserver>
  <regserverlist>
    <count></count>
    <regserver>
      <distributor></distributor>
      <servername></servername>
      <domain></domain>
    </regserver>
    <regserver>
      <distributor></distributor>
      <servername></servername>
      <domain></domain>
    </regserver>
    ...
  </regserverlist>
</teamdrive>
```

All calls, since Registration Server 4.0 will return a `<regserverlist>` tag, which contains a list of the servers found.

Registration Server 4.5.4. will return the URL of each server in the `<domain>` tag, if this information is returned by TDNS. Previous versions of the server only returned the `<domain>` tag for username lookups.

If a username lookup does not find anything, it will throw a “-30100: User unknown” error. In all other cases, the `<count>` tag will be set to zero (“0”).

In addition, only the username lookup will return the single `<regserver>` tag, in addition to the `<regserverlist>` tag. All other calls just return the `<regserverlist>` tag.

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User unknown

### 15.2.4 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>` (see *Identifying Users* (page 123) for details).

The tags `<distributor>`, `<reference>` and `<authid>` were added in version 3.6.0. `<distributor>` determines the authorised Provider, if the caller is authorised for more than one Provider.

The `<userdist>` tag was added in version 4.7.0. This specifies the Provider to search. If not specified, then the Providers searched is determined by the `<onlyownusers>` tag.

If you specify `<onlyownusers>false</onlyownusers>` then you will receive an error if the authorised Provider is not the Default Provider.

When searching for `<authid>` and `<reference>`, you must specify either `<distributor>` or `<userdist>`, unless you the caller is only authorised to access one Provider. This is required because `<authid>` and `<reference>` are not globally unique.

Note that setting `<distributor>` to a value other than your own Provider code is only permitted by the Default Provider.

To retrieve a list of all of 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`.

The tag `<includegroup>` (version 4.0) is optional. The default value is `false`. If the value is `true` then the `<group>` tag is included in the `<user>` block, if the user is a member of a group. In addition, the `<licenselist>` block will also include the user's group license.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>searchuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <email></email>
  <userdist></userdist>
  <reference></reference>
  <authid></authid>
  <startid></startid>
  <showdevice>true|false</showdevice>
  <showlicense>true|false</showlicense>
  <onlyownusers>true|false</onlyownusers>
  <licensereference></licensereference>
  <includegroup>true|false</includegroup>
</teamdrive>
```

**Reply:**

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

```

<regversion></regversion>
<userdist>
  <language></language>
  <webportalurl></webportalurl>
</userdist>
<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>
    <clientsettings></clientsettings>
    <keyrepository>true|false</keyrepository>
    <newsletter>true|false</newsletter>
    <emailbounced>true|false</emailbounced>
    <webportal>true|false</webportal>
    <licensekey></licensekey>
    <licensereference></licensereference>
    <featurevalue></featurevalue>
    <licensestatus></licensestatus>
    <group>
      <distributor></distributor>
      <groupname></groupname>
      <groupreference></groupreference>
      <manager></manager>
      <manageremail></manageremail>
      <groupcreated></groupcreated>
      <groupmodified></groupmodified>
      <groupdepot>
        <depotname></depotname>
        <depotreference></depotreference>
        <hosturl></hosturl>
        <depotid></depotid>
        <globalid></globalid>
        <username></username>
        <accountkey></accountkey>
        <accountreference></accountreference>
        <contractnumber></contractnumber>
        <storagelimit></storagelimit>
        <transferlimit></transferlimit>
        <created></created>
        <etl></etl>
        <status></status>
        <storageused></storageused>
        <transferused></transferused>
      </groupdepot>
      <licensekey></licensekey>
      <licensereference></licensereference>
      <clientsettings></clientsettings>
    </group>
  </user>
</userlist>
<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>
        <isgroup>true|false</isgroup>
        <licenseemail></licenseemail>
    </license>
    <license>...</license>
    ...
</licenselist>
<devicelist>
    <device>
        <deviceid></deviceid>
        <status></status>
        <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 `<group>` tag (version 4.0) is included if `<includegroup>` is `true`, and the user is a member of a group. The `<group>` tag fields are described in the [getgroupdata](#) (page 232) call. A `<memberlist>` block is only included in the `<group>` tag returned by the “`getgroupdata`” call.

The tags `<licensekey>` (version 3.5.10), `<licensereference>` (version 3.6.3), `<featurevalue>` and `<licensestatus>` (version 3.5.9) return 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 162) call (except `<userlist>` tag is not included).

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 127) call.

The `<isgroup>` tag in the `<license>` block is new in version 4.0. This indicates if the license belongs to the user's group. Group licenses are only included if `<includegroup>` is set to `true`.

In version 4.0, the `<licensekey>`, `<licensereference>` (added in version 3.6.3) and `<feature>` tags in the `<device>` block are deprecated and will be removed in a future version.

## 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

### 15.2.5 getuserdata

Get the data associated with a user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>`, `<authid>` or `<activationcode>` (see [Identifying Users](#) (page 123) for details).

`<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.

Use the optional `<settings>` tag to retrieve Registration Server and Provider settings, by specifying a comma separated list of settings names (Registration Server 4.5.1 or later). If included the result will contain a `<settings>` block with the names (as tags) and values of the settings (see [getsettings](#) (page 127) for further details).

The `<includeaccounts>` and `<includegroups>` tag were added in version 4.0. Both are optional, and are set to `true` by default.

If `<includeaccounts>` is `true` then the accounts that the user belongs to are included in the `<accountdata>` block.

If `<includegroups>` is `true` then the `<group>` tag is included in the `<userdata>` block, if the user is a member of a group. In this case, the user's license and depot associated with the user's group will also be included in the `<licensedata>` and `<depotdata>` blocks. All the groups the user belongs to are also included in the `<groupdata>`. `<includegroups>` also effects the license returned in the `<license>` block in `<userdata>` (see the description of the `<license>` block below).

The tags `<includeinusedepots>`, `<includeowneddepots>` and `<includegroupdepot>`, indicate what depots of the user should be returned. If `<includeinusedepots>` is set to `true` (which is the default), then all depots that are in use by the user are returned. If `<includeowneddepots>` is set to `true` (false by default), then the result includes all depots owned by the user. If `<includegroupdepot>` is set to `true` (which is the default), then the result will include the user's account and group depots, if there are any.

These tags are new to Registration Server version 4.0. Previously this call automatically returned all depots in use by the user. There were no options to change this behaviour.

The `<sendmail>` (new in version 4.0) tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getuserdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
  <licensereference></licensereference>
  <settings>RegServerName,ClientSettings,CLIENT_SETTINGS,..</settings>
  <includeaccounts>true|false</includeaccounts>
  <includegroups>true|false</includegroups>
  <includeinusedepots>true|false</includeinusedepots>
  <includeowneddepots>true|false</includeowneddepots>
  <includegroupdepot>true|false</includegroupdepot>
  <sendmail>true|false</sendmail>
  <origin></origin>
</teamdrive>
```

**Reply:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <settings>
    <RegServerName>...</RegServerName>
    <ClientSettings>...</ClientSettings>
    <CLIENT_SETTINGS>...</CLIENT_SETTINGS>
    ...
  </settings>
  <userdata>
    <userid></userid>
    <username></username>
    <email></email>
    <reference></reference>
    <department></department>
    <language></language>
    <distributor></distributor>
    <usercreated></usercreated>
    <status></status>
    <clientsettings></clientsettings>
    <keyrepository>true|false</keyrepository>
    <newsletter>true|false</newsletter>
    <emailbounced>true|false</emailbounced>
    <webportal>true|false</webportal>
    <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>
      <isgroup></isgroup>
      <licenseemail></licenseemail>
```

```

        <language></language>
    </license>
    <group>
        <distributor></distributor>
        <groupname></groupname>
        <groupreference></groupreference>
        <manager></manager>
        <manageremail></manageremail>
        <groupcreated></groupcreated>
        <groupmodified></groupmodified>
        <groupdepot>
            <depotname></depotname>
            <depotreference></depotreference>
            <hosturl></hosturl>
            <depotid></depotid>
            <globalid></globalid>
            <username></username>
            <accountkey></accountkey>
            <accountreference></accountreference>
            <contractnumber></contractnumber>
            <storagelimit></storagelimit>
            <transferlimit></transferlimit>
            <created></created>
            <etl></etl>
            <status></status>
            <storageused></storageused>
            <transferused></transferused>
        </groupdepot>
        <licensekey></licensekey>
        <licensereference></licensereference>
        <clientsettings></clientsettings>
    </group>
</userdata>
<accountdata>
    <account>
        <distributor></distributor>
        <accountkey></accountkey>
        <accountreference></accountreference>
        <created></created>
        <clientsettings></clientsettings>
        <privileges></privileges>
        <jointime></jointime>
    </account>
    <account>...</account>
    ...
</accountdata>
<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>
        <isgroup></isgroup>
        <licenseemail></licenseemail>
    </license>

```

```
        </license>
        <license>...</license>
        ...
    </licensedata>
    <depotdata>
        <count></count>
        <depot>
            <depotname></depotname>
            <depotreference></depotreference>
            <hosturl></hosturl>
            <depotid></depotid>
            <globalid></globalid>
            <username></username>
            <accountkey></accountkey>
            <accountreference></accountreference>
            <contractnumber></contractnumber>
            <storagelimit></storagelimit>
            <transferlimit></transferlimit>
            <created></created>
            <etl></etl>
            <status></status>
            <storageused></storageused>
            <transferused></transferused>
            <isowner></isowner>
            <isdefault></isdefault>
            <iscloud></iscloud>
            <isaccount></isaccount>
            <isgroup></isgroup>
        </depot>
        <depot>...</depot>
        ...
    </depotdata>
    <groupdata>
        <group>
            <groupname></groupname>
            <groupreference></groupreference>
            <memberstate></memberstate>
            <manager></manager>
            <manageremail></manageremail>
        </group>
        <group>...</group>
        ...
    </groupdata>
</teamdrive>
```

The `<userdata>` block is identical to that returned by the [loginuser](#) (page 127) call.

The `<license>` block in `<userdata>` was added in Registration Server version 4.0. It contains the details of the license assigned to the user. This will be the group license if the user is a member of a group with an assigned license, and `<includegroups>` is true. In this case `<isgroup>` will be set to true.

Note that this license may not be one of the licenses listed in the `<licensedata>` block, which contains all the licenses owned by the user.

The `<license>` blocks in `<userdata>` and `<licensedata>` is identical to that returned by the [getlicense-data](#) (page 162) call (except `<userlist>` tag is not included).

The valid values for `<status>` in `<userdata>` include: `todelete`, `disabled`, `inactive` and `activated`.

The `<group>` tag (version 4.0) is included if `<includegroups>` is true, and the user is a member of a group. The `<group>` tag fields are described in the [getgroupdata](#) (page 232) call. A `<memberlist>` block is only included in the `<group>` tag returned by the “getgroupdata” 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 `<isgroup>` tag in the `<license>` block is new in version 4.0. This tag has the value `true` if the license belongs to the user's group. If `<includegroups>` is `false` then group licenses will not be returned in the result.

The `<depotdata>` block contain a number of `<depot>` blocks. The number of of depots in the block is specified in the `<count>` tag.

In this block, `<isowner>` (version 4.0) is set to `true` if the user is the owner of the depot.

In addition, the `<isdefault>` tag is set to `true` if the depot is the user's default depot. There can only be one default depot. The default depot is the depot that was created or assigned to the user automatically when the user is first registered.

`<iscloud>` (version 4.0) in the `<depotdata>` block is set to `true` if this is the user's "selected depot". In general this is the user's group depot if the user is a member of group with a depot, or the Account level selected depot if this exists. Otherwise this is the depot that has been selected on the User level. If the user has no selected depot, then the user's default depot is selected.

`<isaccount>` (version 4.0) in the `<depotdata>` block is set to `true` if this is the depot selected at the Account level. If `<includegroups>` is `false` then account depots will not be returned in the result.

`<isgroup>` (version 4.0) in the `<depotdata>` block is set to `true` if this is the user's Group Depot, which means the depot belongs to the user's group. If `<includegroups>` is `false` then group depot will not be returned in the result.

The `<status>` tag contains one of the following: `to-be-deleted`, `deleted`, `delete-on-server`, `enabled ```, ```disabled`.

The `<accountdata>` block is new in version 4.0. This block is only included if `<includeaccounts>` is `true`. It contains a list of `<account>` blocks with the following fields:

- `<distributor>`: The Provider Code of the account.
- `<accountkey>`: The unique account identifier.
- `<accountreference>`: A unique external reference to the account. This field may be empty.
- `<created>`: The time the account was created.
- `<clientsettings>`: The client settings at the Account level.
- `<privileges>`: The privilege level and status of the user in the account. This is a comma separated list of the following:
  - `member`: the user is regular member of the account.
  - `manager`: the user is a manager of the account.
- `<jointime>`: The time the user became a member of the account.

The `<groupdata>` block is new in version 4.0. This block is only included if `<includegroups>` is `true`. It contains a list of `<group>` blocks with the following fields:

- `<groupname>`: The name or title of the group. The name is not unique and is used for display purposes.
- `<groupname>`: The global unique name of the group.
- `<memberstate>`: This is the state of the user's membership in the group:
  - `member`: the user is a member of the group
  - `invited-as-member`: the user has been invited to join the group as a member
  - `membership-rejected`: the user has refused membership of the group
  - `friend`: the user is a friend of the group
  - `invited-as-friend`: the user has been invited to join the group as a friend

- `friendship-rejected`: the user has refused friendship of the group
- `manager`: the user is the manager of the group
- `<manager>`: The username of the manager of the group.
- `<manageremail>`: The email address of the manager of the group.

Note that this field is a comma separated list of states because a number of state combinations are possible, for example: “member,manager” and “friend,invited-as-member”.

### Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User unknown
- **-30106**: Activation code is invalid
- **-30004**: *Redirect due to user belonging to another Provider* (page 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30127**: License with reference already exists

### 15.2.6 registeruser

Create a new user.

The Provider setting `LOGIN/REG_NAME_COMPLEXITY` (see [REG\\_NAME\\_COMPLEXITY](#) (page 107)) 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. As documented in [USER\\_IDENTIFICATION\\_METHOD](#) (page 108). 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.

A password must be specified using the `<setpassword>` tag unless `<setpassword>true</setpassword>` is used, see below.

The `<newsletter>` specifies if the user wishes to receive the TeamDrive newsletter. This value is `false` by default.

The user will get an activation email sent to their email address. You can change this behaviour using the `<sendmail>` tag as described below.

The user 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 104)) 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 automatically created. If `<licensereference>` contains a value, this will be assigned to the newly created default license.

The `<featurevalue>` tag is optional (added in version 4.0). If specified the features are added to the default license of the user, if the default license does not already exist.

`<featurevalue>` is a comma separated list of the following values: `webdavs`, `personal`, `professional`, `restricted`, `banner`, `secureoffice`, `inbox` and `agent`. The integer values of the features added together may be specified in place of the text values.

If `<featurevalue>` is not specified, then a default license will be generated with the features specified by `LICENSE/DEFAULT_ACCOUNT_FEATURE` if an account has been specified for the user (see `<accountkey>` or `<accountreference>` below), otherwise the `LICENSE/DEFAULT_FREE_FEATURE` determines the features of the license (see [DEFAULT\\_FREE\\_FEATURE](#) (page 102) for details).

The `<accountkey>` or `<accountreference>` tags (version 4.0 or later) specify that the user must be added to an account. In this case, the `<accountprivileges>` tag determines the privilege level of the user in the account. This can be either member or manager.

The `<groupname>` tag (version 4.7 or later) allows you to add the user to a group on creation. The details of the group are then returned in `<group>` tag in the reply.

In version 4.0 or later you can specify user-level client settings in the `<clientsettings>` tag. These settings are appended to the user's Client Settings as specified by the `CLIENT_SETTINGS` Provider Setting (see [CLIENT\\_SETTINGS](#) (page 93)). The user-level Client Settings take priority over group-level settings (see [create-group](#) (page 221) API call) and the Provider values.

The `<activate>`, `<setpassword>`, `<sendmail>` and `<origin>` tag are all new in Registration Server version 4.0.

The `<activate>` tag indicates whether the new user should be activated automatically or not. If `true` the user is automatically activated, if `false` the user must be activated manually. If the tag is omitted the default value depends on the value of the `<sendmail>` tag. If `<sendmail>` is `false` (or `false` by default) then the user is automatically activated.

If the `<setpassword>` tag (default is `false`) is set to `true` then the values of `<activate>` and `<sendmail>` are both ignored. In this case the user's account is not automatically activated, and an email using the **activationsetpassword** template is sent to the user. This email contains a link to the **set-password** HTML template, which allows the user to set his password, and activate their user account.

The `<changeuser>` tag (new in version 4.0) specifies the username of the user that is making the change. Note that this change is required if the `[[ORIGIN-*]]` email variables are to be valid (not empty).

The `<sendmail>` tag indicates whether the user receives an email due to registration or not. If the user was automatically activated, then the user will be sent a **registrationnotify** email. If the user is required to activate manually a **activationlink** email will be sent. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

See [The <origin> tag](#) (page 124) for details on the `<origin>` tag.

The `<message-text>` tag was added in Registration Server version 4.5. When specified, the `[[MESSAGE-TEXT]]` template variable can be used in any email sent by this API call. An example of this is available in the `web-activationsetpassword` template.

The `<sendcc>` tag (new in version 4.5), indicates whether the email sent should be "CC'ed" the user making the change (set by the `<changeuser>` tag). By default this value is `false`.

Starting with Registration Server 4.5 you can assign a depot to a user when the user is registered. You can also elect to prevent the addition of a default depot to the user.

The `<nodedepot>`, when set to `true` indicates that no default depot should be created or assigned to the user. By default this value is `false`, which means the settings `PROVIDER_DEPOT`, `HAS_DEFAULT_DEPOT`,

API\_CREATE\_DEFAULT\_DEPOT and the default depot setting on the account level determine whether the user is assigned a depot on registration.

As another alternative to a default depot is to specify a particular depot to be assigned to the user. In this case the depot is identified using the <hosturl> and <depotid> tags. If the depot is unknown to the Registration Server it will be fetched from the Host Server.

If the depot provided is new to the Registration Server, then <depotreference> is set as external reference to the depot.

The <shopreference> tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>registeruser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useremail></useremail>
  <password></password>
  <language></language>
  <reference></reference>
  <department></department>
  <shopreference></shopreference>
  <newsletter></newsletter>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <featurevalue></featurevalue>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <accountprivileges>member|manager</accountprivileges>
  <groupname></groupname>
  <clientsettings></clientsettings>
  <activate>true|false</activate>
  <setpassword>true|false</setpassword>
  <changeuser></changeuser>
  <sendmail>true|false</sendmail>
  <origin></origin>
  <messagetext></messagetext>
  <sendcc>true|false</sendcc>
  <nodepot>true|false</nodepot>
  <hosturl></hosturl>
  <depotid></depotid>
  <depotreference></depotreference>
</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>
    <clientsettings></clientsettings>
```

```

<keyrepository>true|false</keyrepository>
<newsletter>true|false</newsletter>
<emailbounced>true|false</emailbounced>
<webportal>true|false</webportal>
<group>
  <distributor></distributor>
  <groupname></groupname>
  <groupreference></groupreference>
  <manager></manager>
  <manageremail></manageremail>
  <groupcreated></groupcreated>
  <groupmodified></groupmodified>
  <groupdepot>
    <depotname></depotname>
    <depotreference></depotreference>
    <hosturl></hosturl>
    <depotid></depotid>
    <globalid></globalid>
    <username></username>
    <accountkey></accountkey>
    <accountreference></accountreference>
    <contractnumber></contractnumber>
    <storagelimit></storagelimit>
    <transferlimit></transferlimit>
    <created></created>
    <etl></etl>
    <status></status>
    <storageused></storageused>
    <transferused></transferused>
  </groupdepot>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <clientsettings></clientsettings>
</group>
</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 127) call.

The `<userdata>` block replaces the `<username>` tag (in the `<teamdrive>` block) which was returned since version 3.5.3. The `<username>` tag is still returned but has been deprecated and will be removed in a future version of the Registration Server.

The `<group>` block is new in version 4.0 and is included if the user is a member of a group. The `<group>` tag fields are described in the [getgroupdata](#) (page 232) call. A `<memberlist>` block is only included in the `<group>` tag returned by the “getgroupdata” call.

## 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:** User with given reference already exists
- **-30004:** *Redirect to Registration Server Download Page* (page 144)
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30213:** License disabled
- **-30212:** License has expired
- **-30211:** License exceeded permitted usage
- **-30130:** Unknown group, Group unknown to this Provider
- **-30132:** Account unknown to this Provider

### 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>
```

## 15.2.7 resendactivation

Will resend the activation mail to the user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>`, `<authid>` or `<activationcode>` (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resendactivation</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
  <origin></origin>
</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 130)
- **-30118**: User already activated

### 15.2.8 activateuser

Activate a user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

The <activationcode> tag is must match the activation code for the device, sent to the user in the activation email. This tag is optional in version 4.5.5 or later of the Registration Server. If omitted, the user is activated without checking the activation code.

Since version 4.5.5, this call will also activate all user devices by default. Set the <includeddevices> tag to false to prevent the call from activating devices.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activateuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
  <includeddevices>true|false</includeddevices>
</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 130)
- **-30106:** Wrong activation code

### 15.2.9 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> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deactivateuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail

### 15.2.10 disableuser

Disable the user. This function is available since version 3.5.0.

When disabled, a user is no longer able to access user information using the TeamDrive client or the Registration Server API. The user cannot re-enable himself. Re-enabling the user can only be performed using the “enableuser” API function.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>disableuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
</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 130)

### 15.2.11 enableuser

Enable a disabled 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> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>enableuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
</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 130)

### 15.2.12 activateclient

Activate a TeamDrive Client installation.

If a user registers using the TeamDrive client, 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>
  <distributor></distributor>
  <activationcode></activationcode>
</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

### 15.2.13 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 “change password” 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 *change password* (page 150) API call or via the user’s Client.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see [Identifying Users](#) (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>sendpassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail

### 15.2.14 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> (see [Identifying Users](#) (page 123) for details).

The <sendmail> (new in version 4.0) tag indicates whether the user receives an email or not. If the user is using external authentication then he will be sent a **passwd-reset** email, otherwise a **passwd-invalidated** email. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The <origin> tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resetpassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail

### 15.2.15 changepassword

Change a user's password.

<tmppassword> must contain the temporary password that was emailed to the after the [sendpassword](#) (page 148) 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> (see [Identifying Users](#) (page 123) for details).

The <sendmail> (new in version 4.0) tag indicates whether the user receives a **passwd-changed** email or not. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The <origin> tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changepassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
```

```

    <authid></authid>
    <tmppassword></tmppassword>
    <password></password>
    <sendmail>true|false</sendmail>
    <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30105**: Temporary password does not match
- **-30109**: Password invalid
- **-30137**: Too many failed login attempts

Error -30105 (Temporary password does not match) is returned if the last call to [sendpassword](#) (page 148) (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`.

### 15.2.16 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>` (see [Identifying Users](#) (page 123) for details).

The `<sendmail>` (new in version 4.0) tag indicates whether the user receives a **passwd-changed** email or not. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>updatepassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newpassword></newpassword>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30109:** Password invalid

### 15.2.17 setreference

Set the external reference for a user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setreference</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newreference></newreference>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30127**: User with reference already exists

The error **-30127** will only be returned if the Provider setting EXT\_USER\_REFERENCE\_UNIQUE has been set to True.

### 15.2.18 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> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <command>setdepartment</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <department></department>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail

### 15.2.19 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 155) call does.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setemail</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newemail></newemail>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled

- **-30102:** User not activated by activation mail
- **-30110:** Email invalid
- **-30104:** Email already exists

### 15.2.20 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 156) call (see below).

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see [Identifying Users](#) (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changeemail</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newemail>true|false</newemail>
  <origin></origin>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30110:** Email invalid
- **-30104:** Email already exists

### 15.2.21 confirmnewemail

Confirm the change of email requested by the *changeemail* (page 155) call.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

The <activationcode> tag is required and must match the activation code sent in the email, which was sent to confirm the new email address.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>confirmnewemail</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30106**: Wrong activation code
- **-30104**: Email already exists

### 15.2.22 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> (see *Identifying Users* (page 123) for details).

If <newlanguage> is set to the empty string, then the language of the user will be set to the value of the EMAIL\_DEFAULT\_LANG setting.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changelanguage</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newlanguage></newlanguage>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30115:** Invalid language

**15.2.23 updateuser**

Update various user related fields. This function was added in version 4.0 of the Registration Server.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

The tags <newreference>, <newauthservice> and <newauthid> are used to change the corresponding user identification values.

If <language> is specified as empty, then the language of the user will be set to the value of the EMAIL\_DEFAULT\_LANG setting.

Prior to version 5.0.1, <newlanguage> and <newdepartment> must be used in place of <language> and <department>. These tags are also available in later versions for backwards compatibility.

In version 4.0, only <clientsettings> is optional, and will not be updated if omitted. Omitting any of the other fields when using version 4.0, will remove the current value.

In version 4.1.1 or later of the Registration Server, omitted fields will not be changed.

The <shopreference> tag is new in version 5.0.1.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>updateuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newreference></newreference>
  <newauthservice></newauthservice>
  <newauthid></newauthid>
  <language></language>
  <department></department>
  <shopreference></shopreference>
  <clientsettings></clientsettings>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30127**: User with external reference or authentication ID already exists

The error **-30127** will only be returned when changing the external reference, if the Provider setting `EXT_USER_REFERENCE_UNIQUE` has been set to `True`.

The external authentication ID must always be unique.

### 15.2.24 removeuser

This call will delete the user immediately (as opposed to *deleteuser* (page 160) 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>` (see *Identifying Users* (page 123) for details).

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removeuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <deletelicense>true|false</deletelicense>
  <deletedepot>true|false</deletedepot>
</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 130)
- **-30101:** Wrong password
- **-30137:** Too many failed login attempts

**15.2.25 removedevice**

This call deletes a user's device. The ID of the device must specified in the request.

The list of devices a user posses can be retrieved using [searchuser](#) (page 131).

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see [Identifying Users](#) (page 123) for details).

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removedevice</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <deviceid></deviceid>
</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 130)
- **-30121**: Device not found

### 15.2.26 deleteuser

This call does not delete a user immediately, instead it sends a confirmation email with an “activation code”.

When the user clicks on the link in the email, you are required to call [confirmuserdelete](#) (page 161) in order to actually delete the user.

The [removeuser](#) (page 158) 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> (see [Identifying Users](#) (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deleteuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <origin></origin>
</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 130)

### 15.2.27 confirmuserdelete

Complete the deletion of a user that was initiated by the *deleteuser* (page 160) call.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

The <activationcode> tag is required and must match the activation code sent to the user in the email sent by the *deleteuser* (page 160) 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.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>confirmuserdelete</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <activationcode></activationcode>
  <deletedepot>true|false</deletedepot>
  <deletelicense>true|false</deletelicense>
</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 130)
- **-30101:** Wrong password
- **-30106:** Wrong activation code
- **-30137:** Too many failed login attempts

### 15.2.28 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> (see *Identifying Users* (page 123) for details).

The tag <includegroup> (version 4.0) is optional. The default value is true. If the value is true then the list of licenses returned includes the user's group license if there is one.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getlicensedata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <includegroup>true|false</includegroup>
</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>
      <isgroup>true|false</isgroup>
      <licenseemail></licenseemail>
      <userlist></userlist>
    </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>`).
- `<licensereference>`: An optional external reference that may be used to identify the license.
- `<featurevalue>`: Sum of the numbers as described in `<featuretext>`
- `<featuretext>`: A combination of: banner (1), webdavs (2), personal (4), professional (8), restricted (16), secureoffice (32), agent (64) and inbox (128).
- `<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.
- `<isgroup>` (version 4.0): Set to `true` if the license belongs to the user's group.
- `<licenseemail>`: The email address associated with the license.
- `<userlist>` (version 4.0: A comma separated list of usernames of the users that are using 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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30201**: Unknown license

### 15.2.29 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>` (see *Identifying Users* (page 123) for details).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getdefaultlicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
```

```
<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>
      <userlist></userlist>
    </license>
  </licensedata>
</teamdrive>
```

The `<license>` block is identical to that returned by the [getlicensedata](#) (page 162) 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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail

### 15.2.30 createdepot

Create a depot on the host specified by the `<hosturl>` tag.

The owner of the depot is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

The `<depotreference>` will be stored as an external reference to the depot.

An account may be specified using the `<accountkey>` or `<accountreference>` tag. In this case, the owner must be a manager of the account.

The `<userlist>` tag specified a list of users of the depot. If an account is specified then all users must be members of the account.

By default the depot owner is also added as a user of the depot. Set `<addownerasuser>` to `false` if the owner should not be made a user of depot. Note that you must be a user of a depot in order to create spaces in the depot.

If `<isdefault>` is set to `true` (default is `false`) then the depot is made the default depot of the owner, provided the owner does not already have a default depot.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

The `<shopreference>` tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>createdepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <hosturl></hosturl>
  <depotname></depotname>
  <depotreference></depotreference>
  <shopreference></shopreference>
  <contractnumber></contractnumber>
  <storagelimit></storagelimit>
  <trafficlimit></trafficlimit>
  <userlist></userlist>
  <addownerasuser></addownerasuser>
  <isdefault></isdefault>
  <changeuser></changeuser>
  <changeinfo></changeinfo>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <depotdata>
    <depot>
      <depotname></depotname>
      <depotreference></depotreference>
      <hosturl></hosturl>
      <depotid></depotid>
      <globalid></globalid>
      <username></username>
      <accountkey></accountkey>
      <accountreference></accountreference>
      <contractnumber></contractnumber>
      <storagelimit></storagelimit>
      <transferlimit></transferlimit>
      <created></created>
      <etl></etl>
      <status></status>
      <storageused></storageused>
      <transferused></transferused>
```

```
                <userlist></userlist>
            </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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30132**: Unknown account
- **-30136**: User is not a member/manager of the account
- **-30141**: Cannot create depot, not permitted by license

### 15.2.31 deletedepot

Delete a depot.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
    <command>deletedepot</command>
    <requesttime></requesttime>
    <distributor></distributor>
    <depot></depot>
    <hosturl></hosturl>
    <depotid></depotid>
    <changeuser></changeuser>
    <changeinfo></changeinfo>
    <sendmail>true|false</sendmail>
    <origin></origin>
</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
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.32 updatedepot

Update a depot.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags.

A new owner of the Depot may be specified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

The `<depotname>` tag is optional. If specified, then the name of the depot will be updated.

The `<depotreference>` tag is optional. If specified, then the external reference of the depot will be updated.

The `<contractnumber>` tag is optional. If specified, then the contract number of the depot will be updated.

The `<storagelimit>` tag is optional. If specified, then the storage limit of the depot will be updated.

The `<trafficlemit>` tag is optional. If specified, then the traffic limit of the depot will be updated.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

The `<sendmail>` tag indicates whether the user receives an email or not. If a user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

The `<shopreference>` tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>updatedepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <depotname></depotname>
```

```
<depotreference></depotreference>
<shopreference></shopreference>
<contractnumber></contractnumber>
<storagelimit></storagelimit>
<trafficlimit></trafficlimit>
<changeuser></changeuser>
<changeinfo></changeinfo>
<sendmail>true|false</sendmail>
<origin></origin>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found

### 15.2.33 activatedepot

Activate a depot.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activatedepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <changeuser></changeuser>
```

```
<changeinfo></changeinfo>
</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
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.34 deactivatedepot

Deactivate a depot.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deactivatedepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <changeuser></changeuser>
  <changeinfo></changeinfo>
</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
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found

### 15.2.35 getdefaultdepotdata

This call returns the depot that should be used by default by the user.

In Registration Server 4.0 this is the so-called “cloud depot” which is not necessarily the depot marked as the user’s default (see below).

If the user is a member of a group, and the group has a depot, then the cloud depot is the group depot. Otherwise, if the user has a depot selected on the Account level then this is the cloud depot. If not then this call returns the “selected depot” (which can be set in the Admin Console), specified at the User level.

Finally, if the user has no selected depot, then user’s default depot will be returned, but only if the user’s default depot is in-use.

The tag `<includegroupdepot>` (version 4.0) is optional. The default value is `true`. If the value is `false` then the user’s account and group depots is ignored.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>` (see *Identifying Users* (page 123) for details).

If the tag `<forcecreate>` is set to `true`, then this call will create a new depot if the user does not otherwise have a depot in use, even if `HAS_DEFAULT_DEPOT` and `API_CREATE_DEFAULT_DEPOT` are `false`. Account level settings are also overridden by this tag.

If a depot is created by this call, then the `<depotreference>` value will be stored as external reference to the depot.

The `<sendmail>` (new in version 4.0) tag indicates whether the user receives an email or not. If the user’s depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getdefaultdepotdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <includegroupdepot>true|false</includegroupdepot>
  <forcecreate>true|false</forcecreate>
  <depotreference></depotreference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</teamdrive>
```

Reply:

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

```

        <depotname></depotname>
        <depotreference></depotreference>
        <hosturl></hosturl>
        <depotid></depotid>
        <globalid></globalid>
        <username></username>
        <accountkey></accountkey>
        <accountreference></accountreference>
        <contractnumber></contractnumber>
        <storagelimit></storagelimit>
        <transferlimit></transferlimit>
        <created></created>
        <etl></etl>
        <status></status>
        <storageused></storageused>
        <transferused></transferused>
        <isowner></isowner>
        <isdefault></isdefault>
        <iscloud></iscloud>
        <isaccount></isaccount>
        <isgroup></isgroup>
    </depot>
</depotdata>
</teamdrive>

```

The `<status>` tag contains one of the following: to-be-deleted, deleted, delete-on-server, enabled `` , ``disabled.

`<isowner>` (version 4.0) is set to `true` if the user is the owner of the depot.

`<isdefault>` is set to `true` if the depot is the user's default depot. The default depot is the depot that was created or assigned to the user automatically when the user is first registered.

`<iscloud>` (version 4.0) in the `<depot>` block is set to `true` if this is the user's "cloud depot". In general this is the user's group depot if the user is a member of group with a depot, otherwise the "selected" user depot. If the user has no selected depot, then the user's default depot.

`<isaccount>` (version 4.0) in the `<depotdata>` block is set to `true` if this is the depot selected at the Account level. If `<includegroups>` is `false` then account depots will not be returned in the result.

`<isgroup>` (version 4.0) is set to `true` if this is the user's Group Depot, which means the depot belongs to the user's group. If `<includegroup>` is `false` then group depot will not be returned in the result, and the cloud depot will be either selected or the default depot.

## 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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30107:** No default depot

### 15.2.36 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

### 15.2.37 setdepotforuser

Set a Depot for a user. A user may have multiply Depots, one of which is designated as the default Depot.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>` (see *Identifying Users* (page 123) for details).

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags which identify an existing depot (new in version 4.0). If the depot is unknown on the Registration Server it will be fetched from the Host Server.

If the depot provided is new to the Registration Server, then `<depotreference>` is set as external reference to the depot.

The `<changeuser>` tag (new in version 4.0) specified the username of the user that is making the change.

The `<changeinfo>` tag contains a comment which will be stored in the change history of the depot.

If `<isdefault>` is set to `true`, then the specified depot becomes the default depot of the user.

`<sendtoclient>` has been deprecated in Registration Server 4.0. All changes to a user's Depot configuration are now automatically synchronised with the TeamDrive Client.

The `<sendmail>` (new in version 4.0) tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setdepotforuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <depotreference></depotreference>
  <changeuser></changeuser>
  <changeinfo></changeinfo>
  <isdefault>true|false</isdefault>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.38 removedepotfromuser

Remove the Depot from user. Registration Server 4.0 or later will not return an error if the Depot has already been removed.

If the removed Depot is the default Depot of the user and the user still has other Depots, then the oldest Depot becomes the new default Depot of the user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

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 172) API call.

<sendtoclient> has been deprecated in Registration Server 4.0. All changes to a user's Depot configuration are now automatically synchronised with the TeamDrive Client.

Set the <deletedepot> tag (default false) to true, in order to delete the depot after removing it from the user. Note that this tag will be ignored if the depot is not in use by the user.

The <changeuser> tag (new in version 4.0) specifies the username of the user that is making the change.

The <changeinfo> tag contains a comment which will be stored in the change history of the depot.

The <sendmail> (new in version 4.0) tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The <origin> tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removedepotfromuser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <changeuser></changeuser>
  <changeinfo></changeinfo>
  <deletedepot>true|false</deletedepot>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.39 syncdepotdata

Synchronise the Depot information with the Host Server and the user's TeamDrive Clients.

This call is available in Registration Server 4.0 or later and is used to manually synchronise a user's Depot with the Host Server, and to update the user's Depot configuration on client devices.

This function ensures that the user access list of the Depot on the Host Server is identical to that of the Registration Server. In previous of the Registration Server it was possible that the access list was not synchronised because changes made via the API were not automatically sent to the Host Server, as they are not done in version 4.0.

This function will also send Depot documents to all TeamDrive Client installations of the user that have access to the Depot.

If the `<nosync>` (default `false`) is set to `true` then the function will only send the depot configuration to the users of the specified user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>` (see [Identifying Users](#) (page 123) for details).

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 172) API call.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>syncdepotdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <nosync>true|false</nosync>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted

- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found

#### 15.2.40 getdepotdata

Retrieve current information of a Depot. This call is available in Registration Server 4.0 or later.

The function calls the Host Server to update the Depot information held by the Registration Server if it is older than 30 minutes.

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 172) API call.

If the `<includechanges>` tag (default `false`) is set to `true` then this function will fetch the latest details of the depot from the Host Server. If set to `false`, then the Registration Server will only update the details every 12 hours.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getdepotdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <includechanges>true|false</includechanges>
  <sendmail>true|false</sendmail>
  <origin></origin>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <depotdata>
    <depot>
      <depotname></depotname>
      <depotreference></depotreference>
      <hosturl></hosturl>
      <depotid></depotid>
      <globalid></globalid>
      <username></username>
      <accountkey></accountkey>
      <accountreference></accountreference>
      <contractnumber></contractnumber>
      <storagelimit></storagelimit>
      <transferlimit></transferlimit>
      <created></created>
      <etl></etl>
      <status></status>
      <storageused></storageused>
```

```

        <transferused></transferused>
        <userlist></userlist>
        <lastfetchtime></lastfetchtime>
        <changelist>
            <change>
                <whatchanged></whatchanged>
                <changedate></changedate>
                <changeid></changeid>
                <changedetails></changedetails>
            </change>
            <change>...</change>
            <change>...</change>
        </changelist>
    </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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found

### 15.2.41 getspacedata

Retrieve a list of Spaces belonging to a Depot. This call is available in Registration Server 5.0.1 or later.

The function calls the Host Server to retrieve a list of Spaces of the specified 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 172) API call.

If the `<includedeleted>` tag (default `false`) is set to `true` then list of Spaces will include Spaces marked as deleted on the Host Server.

In order to fetch a “page” of results, set the `<resultlimit>` and `<resultoffset>` tags. `<resultlimit>` specifies the size of the page, and `<resultoffset>` is the row number of the start of the page. The first row is 0 (zero), so if your page size is 10, then the second page begins at offset 10.

The `<origin>` tag is described here: [loginuser](#) (page 127).

Request:

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
    <command>getdepotdata</command>
    <requesttime></requesttime>
    <distributor></distributor>
    <depot></depot>

```

```
<hosturl></hosturl>
<depotid></depotid>
<includedeleted>true|false</includedeleted>
<resultlimit></resultlimit>
<resultoffset></resultoffset>
<origin></origin>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
<regversion></regversion>
<spacedata>
  <etl>true|false</etl>
  <namesincluded>true|false</namesincluded>
  <resultlimit></resultlimit>
  <resultoffset></resultoffset>
  <totalresults></totalresults>
  <space>
    <spaceid></spaceid>
    <name></name>
    <created></created>
    <owneremail></owneremail>
    <owner></owner>
    <status></status>
    <hdrp>true|false</hdrp>
    <lastaccess></lastaccess>
    <storageused></storageused>
    <transferused></transferused>
  </space>
  <space>...</space>
  <space>...</space>
</spacedata>
</teamdrive>
```

The `<totalresults>` tag specifies the total number of Spaces, which may be more than the number of Spaces returned if `<resultlimit>` is less than the total number of Spaces.

`<resultlimit>` and `<resultlimit>` are the same as the values specified in the request.

`<etl>` specifies if the traffic limit is enforced in the spaces.

`<hdrp>` specifies if the Space has a “data retention period”.

`<owner>` is the username of the owner of the Space. If the user is a registered on the Registration Server, then the `<owneremail>` tag contains the email address of the owner.

`<status>` contains the status bits of the Space:

- **BLOCKED = 1** The Space has been disabled by system for maintenance.
- **HOLDING = 2** Space access denied for technical reasons.
- **DELETED = 4** The Space has been deleted by the owner or Provider.
- **DISK\_FULL = 8** The disk limit of the volume has been reached.
- **DEPOT\_FULL = 16** The storage limit of the Depot has been reached (does not stop upload).
- **TRAFFIC\_FULL = 32** The traffic limit for this month has been reached (stops all traffic).
- **DISABLED = 64** General disable flag.
- **READONLY = 128** The Space is set to read-only (no more uploads allowed).
- **STOPPED = 256** The Space has been disabled by the Host Provider.

- **REMOVED = 1024** The Space has no more Host (it has been physically deleted).
- **DEMO\_TIME\_EXP = 2048** This bit means the Host Server/TDPS demo time has expired.
- **RESTORING = 4096** The Space has been temporarily disabled due to restore.
- **FROZEN = 8192** The Depot is so full that uploads are no longer allowed.
- **OVERFLOW = 16384** A large file is being uploaded and will cause volume overflow.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30004**: *Redirect due to user belonging to another Provider* (page 130)
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.42 deletespace

Delete one or more Spaces. This call is available in Registration Server 5.0.1 or later.

The <spaceidlist> tag must contain a comma separated list of Space IDs.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deletespace</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <hosturl></hosturl>
  <depotid></depotid>
  <spaceidlist></spaceidlist>
  <origin></origin>
</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
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found

### 15.2.43 sendinvitation

---

**Note:** This function is deprecated in Registration Server 4.0 and no longer performs the function as previously described. The call will be removed in future versions of the server. Please use the “syncdepotdata” API call instead of this function.

---

Prior to version 4.0 of the Registration Server, this function sent the invitation message provided to the specified user device.

With version 4.0 of the server it is no longer possible to distribute arbitrary Depot documents with this function. Instead, a Depot document must be added to a user using the “setdepotforuser” API call, and then the Depot will be automatically sent to the user’s devices by the server.

In Registration Server 4.0 this call ignores the <invitation> and <type> tags, and simply triggers the Depot distribution mechanism for the specified device.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <userlist></userlist>
  <type></type>
  <invitation></invitation>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail

- **-30111**: Invitation type unknown

#### 15.2.44 setinviteduser

This function is used in the context of the referral program. (see [INVITATION Settings](#) (page 99)). 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.

The `<sendmail>` tag specifies whether the owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setinviteduser</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <inviteduser></inviteduser>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30108**: Invited user can not be found
- **-30209**: Increase user storage failed

### 15.2.45 createlicense

Create a license. You may optionally specify a user or account as owner of the license. The specified user becomes the owner of the license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

If the user has no default license, then the created license will be set to the user's default license.

In version 4.0 or later, the license can be assigned to an account by specifying an account use the <accountkey> or <accountreference> tag.

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, inbox 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 use to identify the license at a later point.
- <email>: This is the **holder email** address of the license. This email address is used to notify the holder of the license of changes to the license, of the license does have an specific owner (a specific user). This value is required if an owner is not specified.
- <language>: This is the language to be used when sending emails to the **holder email** address.
- <contractnumber>: An optional value which may contain any external data relavent 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.
- The <sendmail> tag specifies whether the owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

The <shopreference> tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>createlicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <productname></productname>
  <type></type>
  <featurevalue></featurevalue>
  <limit></limit>
  <licensereference></licensereference>
  <shopreference></shopreference>
```

```

    <contractnumber></contractnumber>
    <contractstatus></contractstatus>
    <contractenddate>YYYY-MM-DD</contractenddate>
    <email></email>
    <language></language>
    <validuntil></validuntil>
    <changeid></changeid>
    <sendmail>true|false</sendmail>
    <origin></origin>
</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
- **-30132:** Unknown account
- **-30100:** User unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30129:** Specify either an account or a user as owner of the license
- **-30203:** Productname unknown
- **-30204:** Type unknown
- **-30205:** Feature unknown
- **-30206:** Limit unknown or invalid
- **-30122:** Invalid date
- **-30110:** Holder email invalid / required
- **-30115:** Invalid language
- **-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`.

### 15.2.46 createlicensewithoutuser

This call has been deprecated in version 4.0. It is now identical to the *createlicense* (page 182) call.

### 15.2.47 assignusertolicense

This call sets the owner of a license to a particular user. If it is the first license to be owned by the user, then it is set to the default license of the user, unless `<isdefault>` (new in Registration Server version 4.0 or later) is set to `false`.

---

**Note:** This function does not set the license used by the user. This is done using *assignlicensetoclient* (page 185).

---

If *createlicensewithoutuser* (page 184) was used, then this call can be used to specify the owner of the license. If the license is already owned by another user or an account, then a **-30211** error will be returned.

You can set the `<removecurrentuser>` tag to `true` (`false` by default) to automatically remove the previous owner (this feature is new in version 4.0).

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>` (see *Identifying Users* (page 123) for details).

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.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>assignusertolicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <removecurrentuser>true|false</removecurrentuser>
  <isdefault>true|false</isdefault>
  <changeid></changeid>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30211:** License already owned by another user or account

### 15.2.48 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 184) 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> (see *Identifying Users* (page 123) for details).

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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30213:** License disabled
- **-30212:** License has expired
- **-30211:** License exceeded permitted usage

### 15.2.49 removeuserfromlicense

Call this function to remove the owner of a license. This is the complement to the [assignusertolicense](#) (page 184) call which sets the owner of a license. The call also removes the license from all groups.

---

**Note:** This call does not change the license usage (see [assignlicensetoclient](#) (page 185) call).

---

The current owner is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Specifying a user is optional in version 3.6.3 or later. If not specified the call will remove the current license owner. If a user is specified and the user is not the owner of the license a **-30201** error is returned. Note that versions 3.6.0 and 3.6.1 incorrectly removed the owner from the license regardless of which user was specified.

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.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removeuserfromlicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30201**: Unknown license

### 15.2.50 deactivatelicence

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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deactivatelicence</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214**: License deleted

Error **-30210**, is no longer returned by version 3.6.3.

### 15.2.51 activatelicence

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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activatelicence</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214**: License deleted

Error **-30210**, is no longer returned by version 3.6.3.

### 15.2.52 deletelicence

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>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
</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

### 15.2.53 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.

`<featurevalue>` is a comma separated list of the following values: `webdavs`, `professional`, `secureoffice`, `agent`, `inbox` and `restricted`. Since version 3.6.3 the integer values of the features added together may be specified in place of the text values. This tag is optional when creating the first license belonging to a user (i.e. the user's default license).

As of version 4.1 the features: `banner` and `personal` are no longer supported and will result in a **-30205** error.

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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>upgradelicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
```

```
<useroremail></useroremail>
<reference></reference>
<authid></authid>
<licensekey></licensekey>
<licensereference></licensereference>
<featurevalue></featurevalue>
<limit></limit>
<changeid></changeid>
<sendmail>true|false</sendmail>
<origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30201**: Unknown license
- **-30214**: 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.

### 15.2.54 upgradedefaultlicense

Upgrade the feature set of a default license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

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, inbox 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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <featurevalue></featurevalue>
  <changeid></changeid>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30205:** Feature unknown

### 15.2.55 downgradelicense

Downgrade a license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

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, inbox 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 downgrad 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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>downgradelicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <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>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30201**: Unknown license
- **-30214**: 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.

### 15.2.56 downgradedefaultlicense

Downgrade the default license of a user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>` (see *Identifying Users* (page 123) for details).

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, inbox 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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <featurevalue></featurevalue>
  <changeid></changeid>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30201:** Unknown license
- **-30214:** License deleted

- -30205: Feature unknown

### 15.2.57 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> (see *Identifying Users* (page 123) for details).

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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <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>
      <isgroup></isgroup>
      <licenseemail></licenseemail>
      <userlist></userlist>
    </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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User 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.

### 15.2.58 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>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <newlicensereference></newlicensereference>
</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
- **-30214:** License deleted
- **-30127:** License with reference already exists

### 15.2.59 removelicence

This call removes the license in use by the user. It undoes the work done by the [assignlicensetoclient](#) (page 185) 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 104)).

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>removelicence</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
</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 130)
- **-30201:** Unknown license
- **-30217:** Cannot remove the user's default license/provider default license

- **-30218:** License is used by the group of the user and cannot be removed

## 15.2.60 cancellicense

Deactivate a license and reduce the number the license usage limit. Use the [deletelicense](#) (page 188) 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 191) 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>` (see [Identifying Users](#) (page 123) for details).

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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>cancellicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <decreaselimit></decreaselimit>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30201**: Unknown license
- **-30214**: 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.

### 15.2.61 setdistributor

Set the Provider of a user.

This function can only be accessed by the Default Provider or by Providers which are managed by a super Provider.

The `<newdistributor>` tag specifies the new Provider of the user.

`<depotreference>` (version 4.0) is used if a new depot is created after the user's Provider has been changed.

`<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`.

The `<sendmail>` (new in version 4.0) tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, then the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The <sendmail> tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setdistributor</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newdistributor></newdistributor>
  <switchdepot>true|false</switchdepot>
  <switchlicense>true|false</switchlicense>
  <depotreference></depotreference>
  <licensereference></licensereference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30127**: License with reference already exists

### 15.2.62 setcapability

Add or remove user capabilities.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid> (see *Identifying Users* (page 123) for details).

<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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30125:** Action must be set or unset
- **-30204:** Unknown capability

### 15.2.63 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> (see *Identifying Users* (page 123) for details).

<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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail

### 15.2.64 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 `<shopreference>`, `<contractstatus>` and `<contractenddate>` tags are new in version 5.0.1.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensecontract</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <contractnumber></contractnumber>
  <shopreference></shopreference>
  <contractstatus></contractstatus>
  <contractenddate>YYYY-MM-DD</contractenddate>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214:** License deleted

### 15.2.65 setlicenseemail

Set the **holder email** address 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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicenseemail</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <email></email>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214**: License deleted
- **-30110**: Holder email invalid / required

### 15.2.66 setlicensefeatures

Set the features 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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicenselanguage</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <language></language>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214:** License deleted
- **-30205:** Feature unknown

**15.2.67 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).

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicenselanguage</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <language></language>
</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
- **-30214**: License deleted
- **-30115**: Invalid language

### 15.2.68 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.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensetype</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <type></type>
  <changeid></changeid>
</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
- **-30214**: 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`.

### 15.2.69 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 owner should be notified via email of the license change (default false). If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensevaliduntil</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <validuntil></validuntil>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214**: License deleted
- **-30122**: Invalid date

### 15.2.70 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).

The `<sendmail>` tag specifies whether the owner should be notified via email of the license change. Note: unlike other API calls that change license data the default value for `<sendmail>` tag is `true`.

If the license has no owner, then the license **holder email** will be used (if this exists). A change notification email is always sent to the Provider of the license using the **License email** address of the Provider.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resetlicensepassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30214**: License deleted

### 15.2.71 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 205) call.

`<password>` is set to the new password.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensepassword</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <tmppassword></tmppassword>
  <password></password>
</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
- **-30214**: License deleted
- **-30101**: Wrong or invalid password

### 15.2.72 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>
  <distributor></distributor>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <password></password>
  <newpassword></newpassword>
</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

- **-30214:** License deleted
- **-30101:** Wrong or invalid password

### 15.2.73 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 97)).

<template> specifies the name of a standard email template.

<language> is optional, if not specified, the language of the user or Provider will by 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>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <template></template>
  <language></language>
  <sender></sender>
  <recipient></recipient>
  <test>true|false</test>
  <origin></origin>
  <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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30216**: Template not found: [template]
- **-30110**: Provider setting <recipient>\_EMAIL is not specified
- **-30110**: No email address specified

### 15.2.74 createaccount

This call creates an account (available since version 4.0).

The <accountcode> tag must contains a 4-character uppercase code. This code is used to generate the account key which has the following format: [provider\_code]-[account\_code]-9999, where 9999 is a random 4-digit number generated by the Registration Server to ensure that the account key is unique.

The account key created by the Registration server is returned in the <accountkey> tag in the reply.

<accountreference> is an optional external identifier of the account. Both the account key and the account reference must be globally unique.

The optional <manager> tag specifies the username of the manager of the account.

The optional <memberlist> tag may be used to specify the first members of the account. <memberlist> is a comma separated list of usernames.

The <shopreference> tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>createaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountcode></accountcode>
  <accountreference></accountreference>
  <department></department>
  <shopreference></shopreference>
  <manager></manager>
  <memberlist></memberlist>
</teamdrive>
```

Reply:

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

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30129**: Account number not specified
- **-30133**: Account key already exists
- **-30127**: Account reference already exists
- **-30135**: User is already a member of another account

### 15.2.75 updateaccount

Update an account (available since version 4.0). The account to be update must specified using the `<accountkey>` or `<accountreference>` tag.

The `clientsettings` tag is required. This specifies the new client level settings for all users of the account.

The `<shopreference>` tag is new in version 5.0.1.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>updateaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <clientsettings></clientsettings>
  <department></department>
  <shopreference></shopreference>
</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
- **-30129**: Account key or reference not specified
- **-30132**: Unknown account

### 15.2.76 deleteaccount

Delete an account (available since version 4.0). The account to be deleted must be specified using the `<accountkey>` or `<accountreference>` tag.

The `<sendmail>` tag indicates whether the user receives an email or not. If a user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The `<sendmail>` tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deleteaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30129**: Account key or reference not specified
- **-30132**: Unknown account

### 15.2.77 addusertoaccount

Add a user to an account (available since version 4.0). The account must be specified using the `<accountkey>` or `<accountreference>` tag.

The user to be added to the account is specified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<accountprivileges>` may either be set to `member`, `manager` or `member,manager` (to add a user as both member and manager).

A user may be a manager of several accounts, but may be a member of only one account.

The tag `<removemembership>` (new in version 4.1.1) specifies whether the user's current account membership should be removed before adding the user as a member to a new account.

By default the value of this value is `false`. In this case, an error occurs if the user is added as a member of an account, and is already a member of some other account.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>addusertoaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <accountprivileges>member|manager</accountprivileges>
  <removemembership>true|false</removemembership>
</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 130)
- **-30129:** Account not specified or incorrect privileges
- **-30132:** Unknown account
- **-30135:** User already a member of another account
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30213:** License disabled
- **-30212:** License has expired
- **-30211:** License exceeded permitted usage

### 15.2.78 inviteusertoaccount

Invite a user to an account via email. This function was added in Registration Server version 4.5. The account must be specified using the `<accountkey>` or `<accountreference>` tag.

The user to be invited to the account is specified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<accountprivileges>` may either be set to `member`, `manager` or `member,manager`. A user may be a manager of several accounts, but may be a member of only one account.

This call will send the “account-manager-invitation” or “account-member-invitation” email template depending on the type of invitation. The user is provided with links in the email to either accept or reject the invitation.

Pending invitations can be removed (before they are accepted) by removing the invited user from the account. If a user rejects an invitation 3 times the user cannot be invited to the account again.

The `<messagetext>` tag can be used to place an invitation text from the inviting user in the email.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>inviteusertoaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <accountprivileges></accountprivileges>
  <messagetext></messagetext>
</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 130)
- **-30129:** Account not specified or incorrect privileges
- **-30132:** Unknown account
- **-30135:** User already a member of another account
- **-30114:** Invited user does not belong to account Provider
- **-30201:** Unknown license
- **-30214:** License deleted
- **-30213:** License disabled
- **-30212:** License has expired
- **-30211:** License exceeded permitted usage

### 15.2.79 removeuserfromaccount

Remove a user from an account (available since version 4.0). The account must be specified using the `<accountkey>` or `<accountreference>` tag.

The user to be removed from the account is specified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

If the user is not a member of the account, no error occurs.

You can specify what account privileges should be removed from the user using the `<accountprivileges>` tag. Any comma separated list of `member`, `manager` or `guest`, is allowed. If not specified, the user is removed from the account completely.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removeuserfromaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <accountprivileges>member|manager|guest</accountprivileges>
</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 130)
- **-30129**: Account not specified
- **-30132**: Unknown account

### 15.2.80 assignaccounttolicense

This call sets the owner of a license to an account. If the license is already owned by a user that is not a member (or manager) of the account, then a **-30211** error will be returned.

If the license is owned by a user of the account, then the license may only be transferred to the account if the license is either not the owner's default license, or the license has a usage limit greater than one.

The account must be specified using the `<accountkey>` or `<accountreference>` tag. The license is specified using `<licensekey>` or `<licensereference>`.

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>assignaccounttolicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
</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
- **-30129**: Account not specified
- **-30132**: Unknown account
- **-30201**: Unknown license
- **-30214**: License deleted
- **-30211**: License already owned by another user or account

### 15.2.81 removeaccountfromlicense

Call this function to remove the ownership of a license by an account. This is the complement to the [assignaccounttolicense](#) (page 214) call which sets the owner of a license to an account. The call also removes the license from all groups.

The account may be specified using the `<accountkey>` or `<accountreference>` tag. Specifying an account is optional. If the account is specified, then the account must be the owner of the account or a **-30201** error is returned.

The license is specified using `<licensekey>` or `<licensereference>`.

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>removeaccountfromlicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
```

```
<accountreference></accountreference>
<licensekey></licensekey>
<licensereference></licensereference>
<changeid></changeid>
</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
- **-30132**: Unknown account
- **-30201**: Unknown license

### 15.2.82 setdepotaccount

Set the account of a depot. A depot may only belong to one account.

The account must be specified using the `<accountkey>` or `<accountreference>` tag.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags which identify an existing depot. If the depot is unknown on the Registration Server it will be fetched from the Host Server.

If the depot provided is new to the Registration Server, then `<depotreference>` is set as external reference to the depot.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The `<sendmail>` tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setdepotaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <depotreference></depotreference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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
- **-30129**: Account not specified
- **-30132**: Unknown account
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found
- **-30134**: Depot already has an account

### 15.2.83 removedepotaccount

Remove the account of a Depot. This function will not return an error if the depot account has already been removed.

The account may be specified using the `<accountkey>` or `<accountreference>` tag. If specified, then the account must match the current account of the depot or a **-30124** error will be returned.

The depot is either identified by the `<depot>` tag, which contains a depot document, or by the `<hosturl>` and `<depotid>` tags which identify an existing depot.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removedepotaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
</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
- **-30132:** Unknown account
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found or does not belong to the account

### 15.2.84 setgroupaccount

Set the account of a group. A group may only belong to one account.

The account must be specified using the <accountkey> or <accountreference> tag.

The group must be specified using the <groupname> tag.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setgroupaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <groupname></groupname>
</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
- **-30130:** Unknown group or the group does not belong to the Provider
- **-30129:** Group or Account reference not specified
- **-30132:** Unknown account
- **-30134:** Depot already has an account

### 15.2.85 removegroupaccount

Remove the account of a group. If the group already has no account, then this call is ignored.

The account may be specified using the <accountkey> or <accountreference> tag. If specified and the account does not match the group's account then error **-30130** is returned.

The group must be specified using the <groupname> tag.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removegroupaccount</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <groupname></groupname>
</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
- **-30132**: Unknown account
- **-30130**: Unknown group or the group does not belong to the account

### 15.2.86 getaccountdata

Return data about an account.

The account must be specified using the `<accountkey>` or `<accountreference>` tag.

Use the optional `<settings>` tag to retrieve Registration Server and Provider settings, by specifying a comma separated list of settings names (Registration Server 4.5.1 or later). If included the result will contains a `<settings>` block with the names (as tags) and values of the settings (see [getsettings](#) (page 127) for further details).

The tag `<includemembers>` is optional, and is `true` by default. If set to `true` the list of account members (`<memberlist>` block) will be included result.

The tag `<includegroups>` is optional, and is `true` by default. If set to `true` the list of account groups (`<grouplist>` block) will be included result.

The tag `<includedepots>` is optional, and is `true` by default. If set to `true` the list of account depots (`<depotlist>` block) will be included result.

The tag `<includelicenses>` is optional, and is `true` by default. If set to `true` the list of account licenses (`<licenselist>` block) will be included result.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getaccountdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <accountkey></accountkey>
  <accountreference></accountreference>
  <settings>ClientSettings,CLIENT_SETTINGS,..</settings>
  <includemembers>true|false</includemembers>
```

```
<includegroups>true|false</includegroups>
<includedepots>true|false</includedepots>
<includelicensees>true|false</includelicensees>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <settings>
    <ClientSettings>...</ClientSettings>
    <CLIENT_SETTINGS>...</CLIENT_SETTINGS>
    ...
  </settings>
  <account>
    <distributor></distributor>
    <accountkey></accountkey>
    <accountreference></accountreference>
    <created></created>
    <clientsettings></clientsettings>
    <accountflags></accountflags>
    <memberlist>
      <member>
        <username></username>
        <email></email>
        <privileges></privileges>
        <username></username>
        <jointime></jointime>
      </member>
      <member>...</member>
      ...
    </memberlist>
    <grouplist>
      <group>
        <distributor></distributor>
        <groupname></groupname>
        <groupreference></groupreference>
        <manager></manager>
        <manageremail></manageremail>
        <groupcreated></groupcreated>
        <groupmodified></groupmodified>
      </group>
      <group>...</group>
      ...
    </grouplist>
    <depotlist>
      <depot>
        <depotname></depotname>
        <depotreference></depotreference>
        <hosturl></hosturl>
        <depotid></depotid>
        <globalid></globalid>
        <username></username>
        <contractnumber></contractnumber>
        <storagelimit></storagelimit>
        <transferlimit></transferlimit>
        <created></created>
        <etl></etl>
        <status></status>
        <storageused></storageused>
        <transferused></transferused>
      </depot>
```

```

        <depot>...</depot>
        ...
    </grouplist>
    <license>
        <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>
            <isgroup>true|false</isgroup>
            <licenseemail></licenseemail>
        </license>
        <license>...</license>
        ...
    </license>
</account>
</teamdrive>

```

The `<privileges>` tag in the `<memberlist>`, is the privilege level and status of the user in the account. This is a comma separated list of the following:

- **member:** the user is regular member of the account.
- **manager:** the user is a manager of the account.
- **guest:** the user has been invited to join a space by a member of the account. This status cannot be combined with other privileges.
- **invited:** the user has been invited to join the account by a manager if the account. This status is combined with **member** and **manager** to indicate with which privilege the user has been invited.
- **invitation-rejected:** the manager invitation has been rejected by the user.

Only users with the **member** status that are not **invited** or **invitation-rejected** are actual members of the account.

The `<accountflags>` tag (included in Registration Server 4.6.4 or later) is a comma separated list of: **2fa-web-only**, **2fa-req**, **enc-web**, **enc-desktop**, **enc-mobile**, **s-pin**, and **s-pin-repo** (see [Super PIN Functionality](#) (page 31)).

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30132:** Unknown account
- **-30129:** Account not specified

### 15.2.87 creategroup

This call creates a group (available since version 4.0).

The manager of the group is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

<groupname> is a globally unique name of the group. The name may consist of the following characters: A-Z, a-z, 0-9, minus (-), underscore (\_) and fullstop (.). The name must include a fullstop. Group names must be unique and may not match any usernames or group names in use.

<groupreference> is an identifier for the group which only unique on the Registration Server of the group. This value is optional.

The <clientsettings> are added to the user's Client Settings as specified by the CLIENT\_SETTINGS Provider Setting (see [CLIENT\\_SETTINGS](#) (page 93)). The group Client Settings take priority over the Provider values.

You may specify a license to be assigned to the group using the <licensekey> or <licensereference> tags. The license must belong to the group manager. When a license is assigned to a group, all members of the group are considered to be using the license whether they have accepted the invitation or not.

A depot may also be set for the group by specifying a depot document in the <depot> tag, or by using the <hosturl> and <depotid> tags to identify an existing depot belonging to the group manager.

If the depot specified with the <depot> tag is new, then <depotreference> will be set as the external reference to the depot.

The <sendmail> tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The <sendmail> tag](#) (page 124).

The <origin> tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>creategroup</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <groupname></groupname>
  <groupreference></groupreference>
  <clientsettings></clientsettings>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <depotreference></depotreference>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30129**: Required parameters not specified
- **-30204**: Invalid group type
- **-30201**: Unknown license or does not belong to the group manager
- **-30214**: License deleted
- **-30213**: License disabled
- **-30212**: License has expired
- **-30127**: Group with given reference already exists
- **-30123**: Depot document/identifiers missing or invalid
- **-30124**: Depot not found or does not belong to the user

### 15.2.88 deletegroup

This call deletes a group (available since version 4.0). The group to be deleted is specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The `<sendmail>` tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deletegroup</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30129**: Group reference not specified
- **-30130**: Unknown group or the group does not belong to the Provider or user

### 15.2.89 inviteusertogroup

Invite a user to a group (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

`<inviteduser>` is the username of the user to be invited.

Invited users are sent an email which contains two links. One link is used to join the group, and the other may be used to reject the invitation. If an invitation is rejected 3 times, a user may no longer be invited.

The `<invitetype>` tag specifies if the user should be invited as a member or a friend. The default is member.

Users can only be a member of one group (but can be a friend of any number of groups). This means that if a user accepts an invitation as a member, then the user will be removed as a member from any other group. In this case, the user's status in that group reverts to `invited-as-member`.

If invited as a member and the group has a license then the user is considered to be using the license even if the user has not accepted the invitation.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>inviteusertogroup</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <inviteduser></inviteduser>
```

```

    <invitetype>member|friend</invitetype>
    <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30108**: Invited user unknown or does not belong to the Provider
- **-30131**: The user cannot be invited because invitation was rejected too many times
- **-30129**: Group reference not specified, Unknown invitation type:, Unknown membership status:

### 15.2.90 removeuserfromgroup

Remove a user from a group (available since version 4.0) or cancel an invitation to a group. The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

`<removeuser>` is the username of the user to be removed.

No error occurs if the user has already been removed from the group, or is not a member of the group. Removing a user from a group will reduce the usage of any license used by the group accordingly.

The number of times a user has rejected invitation to the group is not reset by removing the user from the group.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The `<sendmail>` tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
    <command>removeuserfromgroup</command>
    <requesttime></requesttime>

```

```
<contributor></contributor>
<groupname></groupname>
<username></username>
<useroremail></useroremail>
<reference></reference>
<authid></authid>
<removeuser></removeuser>
<sendmail>true|false</sendmail>
<origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30108**: Removed user unknown or does not belong to the Provider
- **-30129**: Group reference not specified

### 15.2.91 setgrouplicense

Set the license used by users of a group (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

The license to be assigned to the group must be specified using the `<licensekey>` or `<licensereference>` tags. The license must belong to the manager of the group. Setting the group license will automatically remove any previously assigned license.

All members of the group will be counted as users of the license. If the license has insufficient usages available, then an error will occur.

Request:

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

```

    <requesttime></requesttime>
    <distributor></distributor>
    <groupname></groupname>
    <username></username>
    <useroremail></useroremail>
    <reference></reference>
    <authid></authid>
    <licensekey></licensekey>
    <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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30201**: Unknown license or does not belong to the group manager
- **-30214**: License deleted
- **-30213**: License disabled
- **-30212**: License has expired
- **-30211**: License exceeded permitted usage
- **-30129**: Group reference not specified
- **-30212**: Group license may not have an expiry date

### 15.2.92 removegrouplicense

Remove the group license (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

If the group has a license it will be removed from the group. All members of the group revert to using their own default license.

The license is only removed from the group, not from the list of licenses belonging to the manager.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removegrouplicense</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30129**: Group reference not specified

### 15.2.93 setgroupdepot

Set the depot for a group (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

The depot to be set is specified as a depot document using the `<depot>` tag, or by specifying the `<hosturl>` and `<depotid>` tags to identify an existing depot belonging to the group manager.

If the depot specified with the `<depot>` tag is new, then `<depotreference>` will be set as the external reference to the depot.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The `<sendmail>` tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setgroupdepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <depot></depot>
  <depotreference></depotreference>
  <hosturl></hosturl>
  <depotid></depotid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30130:** Unknown group or the group does not belong to the Provider or user
- **-30123:** Depot document/identifiers missing or invalid
- **-30124:** Depot not found or does not belong to depot manager
- **-30129:** Group reference not specified

### 15.2.94 removegroupdepot

Remove the Group Depot (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

The depot is only remove from the group, not from the list of depots belonging to the manager.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: *The `<sendmail>` tag* (page 124).

The `<origin>` tag is new version 4.0 and is described here: *loginuser* (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removegroupdepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30129**: Group reference not specified

### 15.2.95 userjoinedgroup

This call can be used to confirm that a user that has been invited to a group, has accepted the invitation. In other words, it performs the function that would normally be done by the user when he clicks on the “accept link” in the group invitation email.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

If the user was invited as a member of the group then the user will become a member of the group, and if the user was invited as a friend, then the user becomes a friend of the group.

If the user is already a member of the group, this call will not change the user's status in the group.

The `<activationcode>` tag is required and must match the activation code sent in the email inviting the user to join the group.

The `<sendmail>` tag indicates whether the user receives an email or not. If the user's depot configuration changes due to this call, the user will be sent a **depotchanged** email. If the tag is omitted the default depends on a number of factors described here: [The `<sendmail>` tag](#) (page 124).

The `<origin>` tag is new version 4.0 and is described here: [loginuser](#) (page 127).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removegroupdepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <activationcode></activationcode>
  <sendmail>true|false</sendmail>
  <origin></origin>
</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**: Unknown or incorrect activation code
- **-30131**: User cannot join group, already rejected membership
- **-30129**: Group reference not specified
- **-30130**: Unknown group or the group does not belong to the Provider or user

### 15.2.96 setgroupclientsettings

Set the client settings for a group (available since version 4.0). The group must be specified using the `<groupname>` tag.

The group manager may be specified in the request using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`. Error **-30130** will occur if the specified user is not the manager of the group.

The Client Settings specified using the `<clientsettings>` tag are added to the Client Settings for every member of the group. The Client Settings specified by the `CLIENT_SETTINGS` Provider Setting continue to apply but the group setting values take priority.

For a complete list of allowed settings see chapter [Login and Registration Client Settings](#) (page 112)

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setgroupclientsettings</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <clientsettings></clientsettings>
</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 130)
- **-30120**: User has been deleted
- **-30119**: User is disabled
- **-30102**: User not activated by activation mail
- **-30130**: Unknown group or the group does not belong to the Provider or user
- **-30129**: Group reference not specified

### 15.2.97 getgroupdata

Return the details of a group (available since version 4.0). The group must be specified using the <groupname> tag.

The group manager may be specified in the request using one of the following tags: <username>, <useroremail>, <reference> or <authid>. Error **-30130** will occur if the specified user is not the manager of the group.

An error will occur if the group does not belong to the calling Provider or the specified group manager.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getgroupdata</command>
  <requesttime></requesttime>
  <distributor></distributor>
  <groupname></groupname>
  <username></username>
  <useroremail></useroremail>
```

```

    <reference></reference>
    <authid></authid>
</teamdrive>

```

**Reply:**

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <group>
    <distributor></distributor>
    <groupname></groupname>
    <groupreference></groupreference>
    <manager></manager>
    <manageremail></manageremail>
    <groupcreated></groupcreated>
    <groupmodified></groupmodified>
    <groupdepot>
      <depotname></depotname>
      <depotreference></depotreference>
      <hosturl></hosturl>
      <depotid></depotid>
      <globalid></globalid>
      <username></username>
      <accountkey></accountkey>
      <accountreference></accountreference>
      <contractnumber></contractnumber>
      <storagelimit></storagelimit>
      <transferlimit></transferlimit>
      <created></created>
      <etl></etl>
      <status></status>
      <storageused></storageused>
      <transferused></transferused>
    </groupdepot>
    <licensekey></licensekey>
    <licensereference></licensereference>
    <clientsettings></clientsettings>
    <memberlist>
      <member>
        <username></username>
        <email></email>
        <memberstate></memberstate>
        <rejectcount></rejectcount>
        <invitetime></invitetime>
        <modifytime></modifytime>
        <activationcode></activationcode>
      </member>
      <member>
        <username></username>
        <email></email>
        <memberstate></memberstate>
        <rejectcount></rejectcount>
        <invitetime></invitetime>
        <modifytime></modifytime>
        <activationcode></activationcode>
      </member>
      ...
    </memberlist>
  </group>
</teamdrive>

```

The <group> block includes the following fields:

- `<distributor>`: The Provider Code of the group.
- `<groupname>`: Is the globally unique name of the group.
- `<groupreference>`: The Registration Server wide unique identifier of the group. This value is optional.
- `<manager>`: The username of the group manager.
- `<manageremail>`: The email address of the group manager.
- `<groupcreated>`: The date of group creation.
- `<groupmodified>`: The time of the last change to the group.
- `<groupdepot>`: This block contains a reference to the depot of the group. This value is optional.
- `<licensekey>`: The license key of the group license. This value is optional. If the group has a license, all members of the group will use the group license instead of one of their own licenses.
- `<licensereference>`: The license reference of the group license.
- `<clientsettings>`: The Client Setting for the group. These settings are added to the user's Client Settings as specified by the `CLIENT_SETTINGS` Provider Setting (see [CLIENT\\_SETTINGS](#) (page 93)). The group Client Settings take priority over the Provider values.
- `<memberlist>`: A list of members in the group.

The `<groupdepot>` block includes the following fields:

- `<hosturl>`: The Host Server URL of the depot. This value has the form: [http://<domainname>\[:<port>\]](http://<domainname>[:<port>]). Together with the depot ID (`<depotid>` below) this value uniquely and globally identifies a depot.
- `<depotid>`: This is the ID of the depot on the Host Server.

The `<member>` block includes the following fields:

- `<username>`: The username of the group member.
- `<email>`: The email address of the group member.
- `<memberstate>`: This is the state of membership as described in the “getuserdata” call ([getuserdata](#) (page 135)).
- `<rejectcount>`: The number of times the user has refused membership.
- `<invitetime>`: The time the first invitation was sent to the user.
- `<modifytime>`: The last time the membership status changed.
- `<activationcode>`: The activation code is used to identify an invitation in the invitation email sent to the user.

If a group has a license or depot, then the license or depot is only used by users of the group when the membership state is `member`. Users continue to use their own license and depot as in all other states, including: `invited-as-member` and `membership-rejected`.

However, for the purpose of license usage, any state that involves membership is counted, including: `member`, `invited-as-member` and `membership-rejected`. In other words, if a group has a license, then the license must be below its maximum usage in order to invite a user as a member.

### 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 130)

- **-30120:** User has been deleted
- **-30119:** User is disabled
- **-30102:** User not activated by activation mail
- **-30130:** Unknown group or the group does not belong to the Provider or user
- **-30129:** Group reference not specified

## 15.3 Error Codes

The following table lists all API error codes.

Table 15.1: API Error Codes

Error	Code	Comment
ACCESS_DENIED	-30000	Access denied
INVALID_COMMAND	-30001	Invalid Command
INVALID_REQUEST	-30002	Invalid Request
INVALID_XML	-30003	Invalid XML
DISTRIBUTOR_REDIRECT	-30004	Returns a URL reference to a different distributor
USER_UNKNOWN	-30100	User not found
WRONG_PASSWORD	-30101	Wrong password
USER_NOT_ACTIVATED	-30102	User not activated by activation mail
USERNAME_ALREADY_EXISTS	-30103	Username already exists
EMAIL_ALREADY_EXISTS	-30104	Email already exists
TEMP_PASSWORD_NOT_MATCH	-30105	Temporary password does not match
WRONG_ACTIVATION_CODE	-30106	Wrong activation code
NO_DEFAULT_DEPOT	-30107	No Default Depot
USERNAME_INVALID	-30108	Username invalid
PASSWORD_INVALID	-30109	Password invalid
EMAIL_INVALID	-30110	Email invalid
INVITATION_TYPE_UNKNOWN	-30111	Invitation type unknown
INVALID_LOCATION	-30112	Invalid location
TEMP_PASSWORD_EXPIRED	-30113	Temporary password expired
INVALID_DISTRIBUTOR	-30114	Provider not found or invalid
INVALID_LANGUAGE	-30115	Invalid language
SEARCH_STRING_TOSHORT	-30116	Search conditions too short or missing
ACTIVATION_CODE_NOT_FOUND	-30117	Activation code not found
ACCOUNT_ALREADY_ACTIVATED	-30118	Account already activated
ACCOUNT_DISABLED	-30119	Account disabled
ACCOUNT_TODELETE	-30120	Account will be deleted
DEVICE_NOT_FOUND	-30121	Device not found
INVALID_DATE	-30122	Invalid date
DEPOT_INVALID	-30123	Depot invalid
DEPOT_NOT_FOUND	-30124	Depot not found
INVALID_PARAMETER	-30125	Invalid parameter
LOGIN_EXPIRED	-30126	Login expired
DUPLICATE_EXT_REF	-30127	Duplicate external reference
EMAIL_IN_USE_BY_EXT	-30128	Email in use by some other Registration Server
INVALID_INPUT	-30129	Invalid Input
GROUP_UNKNOWN	-30130	Group unknown
INVITE_REJECTED	-30131	Invite Rejected
ACCOUNT_UNKNOWN	-30132	Account unknown
ACCOUNT_EXISTS	-30133	Account exists

Continued on next page

Table 15.1 – continued from previous page

Error	Code	Comment
DEPOT_IN_USE	-30134	Depot in use
ALREADY_MEMBER	-30135	Already member
NOT_A_MEMBER	-30136	Not a member
TOO_MANY_LOGINS	-30137	Login attempts exceeded
CANNOT_REMOVE_USER	-30138	Cannot remove the master user
UPDATE_DATABASE	-30139	Database update required
OBJECT_ALREADY_EXISTS	-30140	Account already has an inbox service
INVALID_LICENSE	-30141	A different type of license is required
UNKNOWN_LICENSE	-30201	Unknown License
LICENSE_UPGRADE_FAILED	-30202	License Upgrade failed
PRODUCTNAME_UNKNOWN	-30203	Productname unknown
TYPE_UNKNOWN	-30204	Type unknown
FEATURE_UNKNOWN	-30205	Feature unknown
LIMIT_UNKNOWN	-30206	Limit unknown
CANCEL_LICENSE_FAILED	-30207	Cancel license failed
LICENSE_DOWNGRADE_FAILED	-30208	Downgrade license failed
INCREASE_SPACE_FAILED	-30209	Increase user storage failed
LICENSE_CHANGE_FAILED	-30210	License change failed
LICENSE_IN_USE	-30211	License belongs to another user
LICENSE_EXPIRED	-30212	License expired
LICENSE_DEACTIVATED	-30213	License deactivated
LICENSE_DELETED	-30214	License deleted
CONFIGURATION_ERROR	-30215	Configuration error
TEMPLATE_UNKNOWN	-30216	Template unknown
USING_DEFAULT_LICENSE	-30217	Default license is in use

## 15.4 User Change Notifications

You can enable user change notifications by setting the Provider setting `API/API_ENABLE_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.

### 15.4.1 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 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. Disabled users cannot be accessed by the TeamDrive client.
- "to-delete" is set in order to schedule a user 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 11). 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`.

### 15.4.2 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 `send_notifications_task` sends notifications that have been delayed for some reason. The task runs every 5 minutes by default.

## 16.1 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 47) 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.

## 16.2 Abbreviations

**PBT** PrimeBase Talk

**SAKH** Server Access Key HTTP for TeamDrive 2.0 Clients

**TDNS** TeamDrive Name Service

**TDPS** TeamDrive Personal Server

**TDRS** TeamDrive Registration Server

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