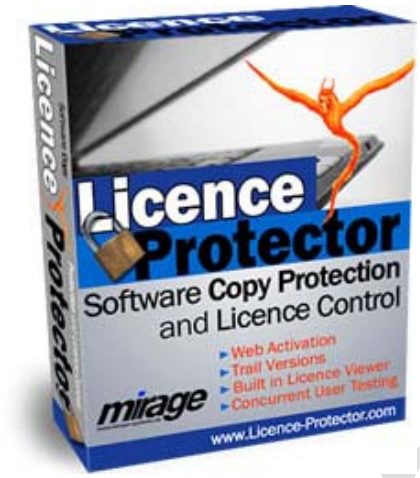
mirage
www.mirage-systems.de

Licence Protector



Web Activation Server Documentation

Licence Protector version: 2.3.2
Date: 05/01/2005

Table of Contents

1 Introduction

1.1 Documentation

This document describes how to set up the Licence Protector Web Activation server, the necessary configuration and the basic implementation of the product. It refers to other documentation which should be read in advance.

- *Licence Protector Developer Documentation* – explains the concepts of Licence Protector, points out how you can use the DLL interface and how you distribute Licence Protector to your customers. We strongly recommend, that you read this documentation

  This documentation names the above mentioned specification with the abbreviation: *Developer Documentation*

- *Licence Protector Web Activation – Step by Step* – provides an overview how to use the *Web Activation tool* and allows testing with a sample application using the demo Web Activation server

- *Licence Protector – Automatic Licence Generator Documentation* – explains how to set up an automated process to create licence files and keys. The Automated Licence Generator is implemented in the Web Activation Server and configuration files from the Automatic Licence Generator are used

1.2 Knowledge Base

You will find a lot of add on information, tips, implementation issues and support for typical error messages within the online *Knowledge Base* at www.Licence-Protector.com, Select Support, then Knowledge Base. Select Licence Protector and type in a search criteria or browse through the chapter *Web Activation*.

Customers with a valid update subscription do have access to additional information within the *Customer Self Service Portal*.

1.3 Abbreviations

The following abbreviations are used:

- LPWAS - Licence Protector Web Activation Server
- ALG – Automatic Licence Generator

1.1. Licence Protector Web Activation - Overview

The Licence Protector Web Activation Server (**LPWAS**) is an enhancement to the standard Licence Protector system.

The Web Activation and registration server is an '*out of the box solution'* to activate a licence and to change licence information **online**. With just 1 DLL call, Licence Protector connects to the registration server and performs the following operations:

- Activate copy protection
- Change a license, e.g. from demo mode to full version mode
- Delete time limitation
- And any other workflow you can do with keys

The Activation Server ensures that each licence can be registered only once. Every transaction is logged into a database (database licence not included).

Optionally, keys can be transmitted to the Activation Server before they are applied to the licence file. It is checked, if the key was already used. This is to ensure, that the key can not be passed on and used multiple times.

The Web Activation service includes 2 different applications

- Activate licence file (module)
- Online Validation of Keys

1.4 Activate a Licence File

During the creating of the licence file you can set a **property per module**, that the module has to be activated online. The activation per module adds additional security when a module is subsequently purchased. Technically not the licence file but the module is activated.

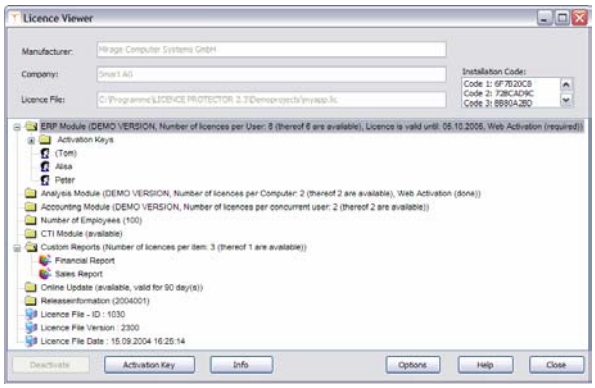
What is done during activation is **defined on the server**. Multiple actions are possible:

- Activate Software Copy Protection. All supported installation codes are sent to the server. The software copy protection status is set in the licence file in a predefined order (e.g. MAC address, hostname, volume ID)
- Generation of multiple Activation Keys, e.g. to activate the main module, delete time limitation and demo status

The server allows extensive configuration settings:

- Frequency, how often a customer can activate a licence. Therewith you can e.g. give the customer a leased licence which he has to activate on a monthly base
- To identify a customer, a unique ID is necessary. This ID can be selected freely and could be the licence file ID (serial number of the licence file), E-Mail address, invoice number or hardware criteria like MAC address
- A predefined positive list can be imported to define, which user can activate a licence (unique ID is necessary). Herewith you can implement advanced functionality like only customers with a valid invoice number can activate a licence

Licence Viewer or own interface	**Transmit unique ID and additional data** **Receive Installation Code and Keys**	**Web Activation Server** (predefined Workflow) – checks the unique criterion (licence file ID, key, e-mail, installation code) and allows only a predefined number of registrations – normally 1



B3F58883

1.5 Validate an Activation Key

If Activation Keys are used, there are several possibilities to prevent misuse:

- File ID or Installation code can be stored within the Activation Key
- Validate Activation Keys via the Web server

1.6 Store File ID / Installation Code within the Activation Key

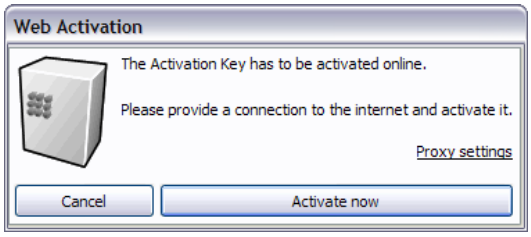
With this feature, the Activation Key can only be used for the specified licence file. However you need the File ID / Installation code of the customer while creating a licence file.

1.7 Validate Activation Keys online via Web Server

During the creating of an Activation Key it is defined, that the key has to be validated **online before** it can be applied to the licence file. The Activation Key can be applied via the Licence Viewer or your own interface and is then sent to the activation server. There it is validated, whether it was already used or not and stored into the database. This ensures that the Activation Key can never be used twice.

Validate a Key User types in a key via Licence Viewer or within your application with DLL method *ApplyActivationKey*	**Transmit key** **Receive status key already**	**Web Activation Server** Checks if the key is already in the database and stores the key with

used or not additional information



2 Installation

2.1 System Requirements

To run a LPWAS you need the following components:

- Microsoft Internet Information (IIS) Server Version 5 or higher. The FrontPage Extensions are not necessary.
  This server has to be reached from all your customers using the LPWAS features
- Microsoft .NET framework 1.1
- Microsoft SQL Server 2000 (or higher) or Microsoft MSDE
- Right to start an executable on the server

2.2 Install LPWAS

2.2.1 Prepare the IIS

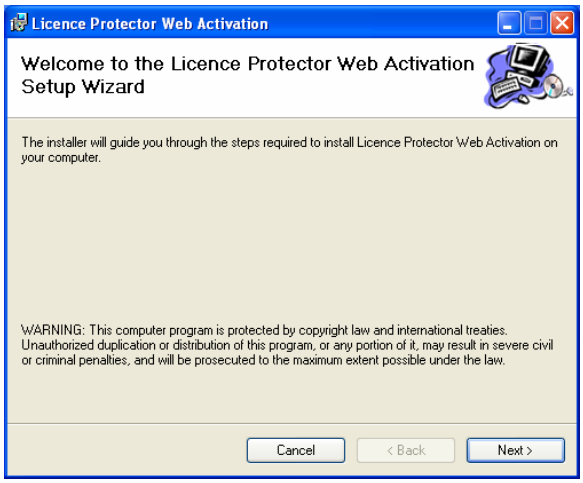
Install the ASP.NET facility inside the IIS. Do this by executing `Aspnet_regiis.exe -i` from the command prompt. This utility can be found inside the Windows folder in the .NET 1.1 subdirectory:

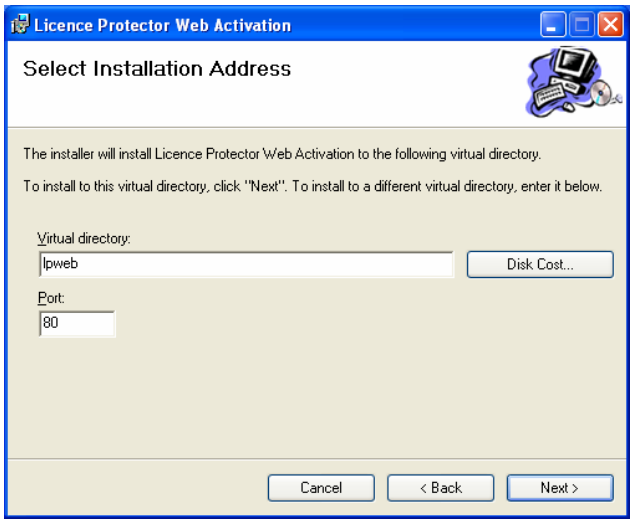
```
C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322>aspnet_regiis.exe -i
Start installing ASP.NET (1.1.4322.0).
Finished installing ASP.NET (1.1.4322.0).
```

2.2.2 Install the LPWAS Application

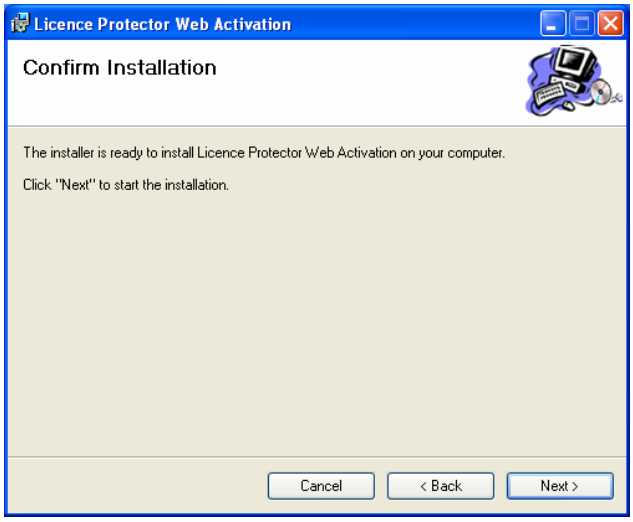
The LPWAS is bundled in the LPWAsetup.msi file. You have to execute it to let the files be extracted and be placed correctly.

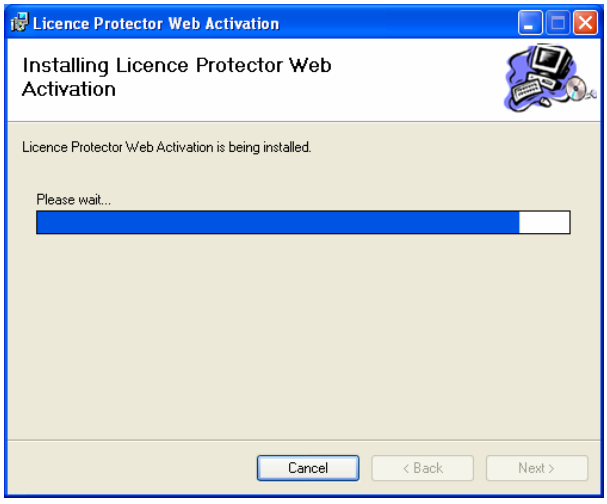
To install simply execute LPWAsetup.msi and press Next on all screens:

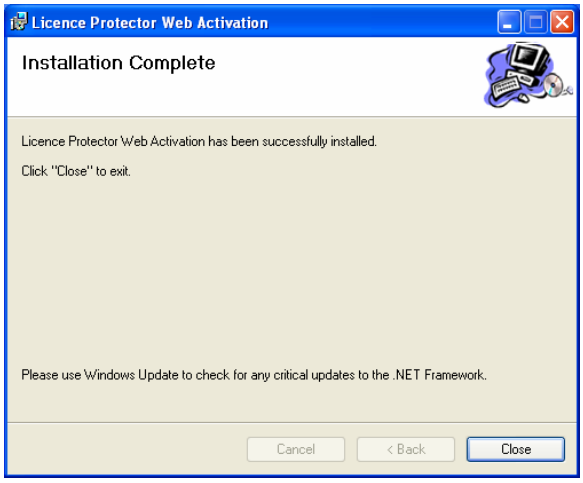




You can change the Virtual directory or the port. Remember your input to adapt the installation properly. We recommend using the name “lpweb” and port 80.







Now LPWAS is installed. Continue with creating and setting up the database.

The installation files are copied to

```
\inetpub\wwwroot\lpweb
```

Note:
Replace the file LicProtector.lic that was installed by LPWAsetup.msi with that file you have received when you bought Licence Protector. The file is on the folder *\inetpub\wwwroot\lpweb\bin\alg*

2.2.3 Necessary rights for the ASP Service

The ASP application works with an IIS-process 'user'. This user must have full read and write. The user name is **IWAM_Machine Name**. These rights should be given to the complete lpweb folder with all subfolders. The same is for the user **IUSR_Machine name** and user **ASP.NET** access.

- Choose the lpweb folder
- Right-click and choose Properties
- On the "Directory Security" tab you can edit the access rights
- Set all rights to full access
- Click Advanced
- Select Inherit from parent the permission entries that apply to child objects

If you get error 3013 during activation then include the user 'everyone' with the same rights. If this does not solve the error then include the user 'everyone' with the same rights.

2.2.4 Create the database and set up table structure

We provide 2 ways to create the LPWAS database:

- The osql command line utility
- The SQL Server Enterprise Manager (only available if you have SQL Server installed)

Use the osql utility

Step1: Open a command prompt window (e.g. Run → `cmd`)
Move to the db subdirectory. For example:

```
C:
Cd \inetpub\wwwroot\lpweb\db
```

Step2: Now start the osql:

```
osql -S localhost -U sa -P secret
```

The parameter –S defines the database server. If you use a database server on your local machine then use *localhost*.

The parameter –U defines the database user. You can use the standard system administrator *sa*.
The parameter –P defines the password for the user of parameter –U (in our example it is the word „secret“). If you have no password set (default password for sa is empty) then use –P without a password following.

We strongly recommend setting passwords for all your database users!
If you use no –P parameter then osql requests the password. It can't be seen on that that way.

Step 3: After logging on to osql you see the prompt `1>`
Now create the database:
`Create database LPWeb23` <press Return>
Type `go` and press Return.
Now you see that the database is created.

Step 4: Leave osql by typing `exit`.

```
C:\Inetpub\wwwroot\lpweb\db>osql -S localhost -U sa -P secret
1> create database LPWeb23
2> go
The CREATE DATABASE process is allocating 0.63 MB on disk 'LPWeb23'.
The CREATE DATABASE process is allocating 0.49 MB on disk 'LPWeb23_log'.
1> exit

C:\Inetpub\wwwroot\lpweb\db>
```

Step 5: Next you have to create the table structure for LPWAS.
Enter the following:

```
osql -S localhost -U sa -P secret -d LPWeb23 -i create.sql
```

The Parameters –S, -U, -P are the same as above.

The Parameter –d defines the name of the database. You have to use the name that your newly created database has received in step 3.
The Parameter –i defines the name of the database script. If you have moved your command prompt to the `db` directory you do not need to provide a path. Otherwise you have to enter the full path e.g. …`-i c:\inetpub\wwwroot\lpweb\db\create.sql`

```
C:\Inetpub\wwwroot\lpweb\db>osql -S localhost -U sa -P secret -d LPWeb23 -i crea
te.sql
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 15> 16> 17> 18> 19> 20> 21> 22> 2
3> 24> 25> 26> 27> 28> 29> 30> 31> 32> 33> 34> 35> 1> 2> 3> 4> 5> 6> 7> 8> 9> 10
> 11> 12> 13> 14> 15> 16> 17> 18> 19> 20> 1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12>
 13> 14> 15> 1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 1> 2> 3> 4> 5> 6> 7>
 8> 9> 10> 11> 12> 13> 1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 1> 2> 3> 4
> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 15> 1> 2> 3> 4> 5> 6> 7> 8> 1> 2> 3> 4> 1>
2> 3> 4> 1> 2> 3> 4> 1> 2> 3> 4> 1> 2> 3> 4> 1> 2> 3> 4> 1> 2> 3> 4> 1> 2> 3> 4>
 1> 2> 3> 4> 5> 6> 7> 8> 9> 1> 2> 3> 4> 5> 1> 2> 3> 4> 5> 1> 2> 3> 4> 5> 1> 2> 3
> 4> 5> 1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 15> 16> 17> (1 row affect
ed)
1>
C:\Inetpub\wwwroot\lpweb\db>
```

Now the database setup is complete. Continue with the customization of the config-files.

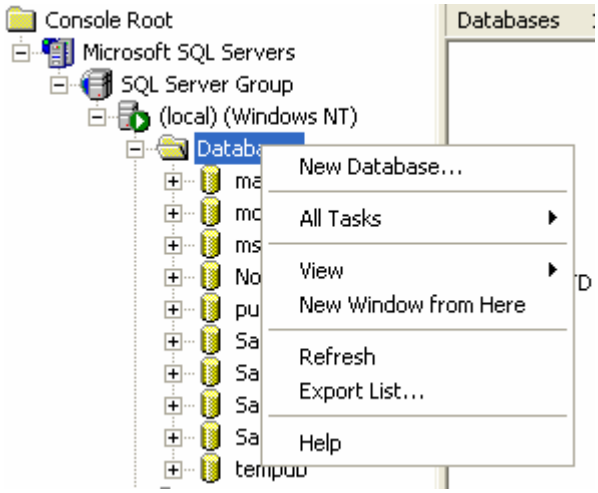
Use SQL Server Enterprise Manager

Note that thie tool only is availlable if you have SQL Server installed. If you use MSDE there is no Enterprise Manager. In that case you have to use osql instead (see above).

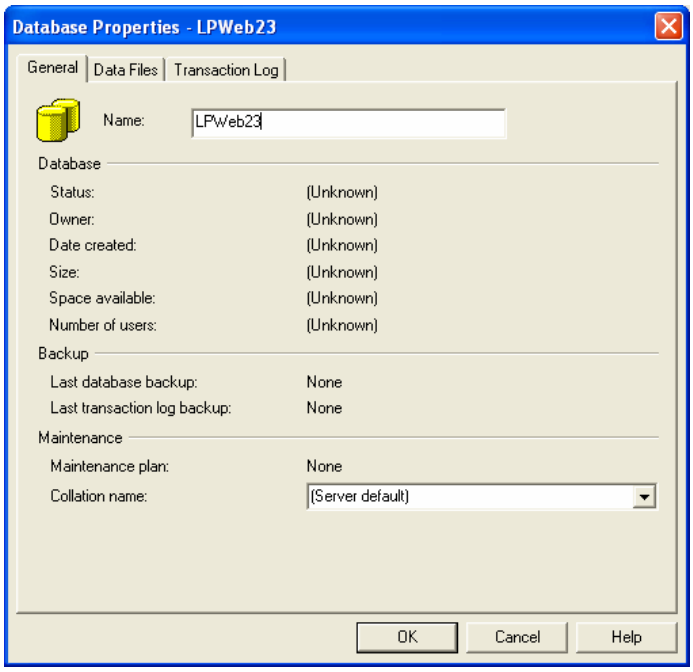
Step 1: Open Enterprise Manager

Step 2: Expand the tree on the left side:
Console Root → Microsoft SQL Servers → SQL Server Group → your Server → Databases

Step 3: Right click on Databases and choose *New Database*.



Step 4: Enter the name e.g. *LPWeb23* and press the *Ok* button. Now the database is created.



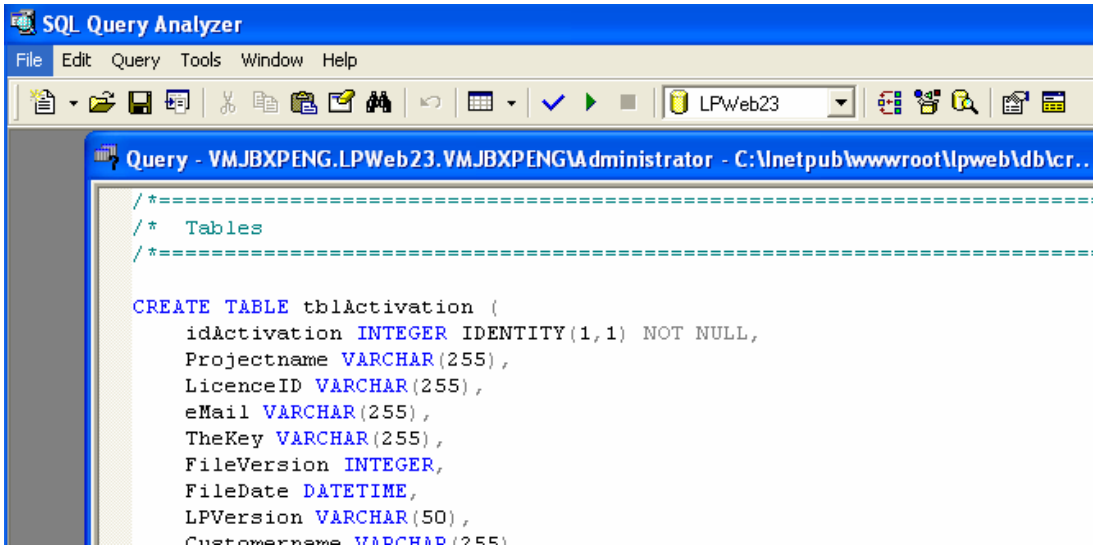
Step 5: Double-Click on the newly created database. From menu above choose „*Tools→SQL Query Analyzer*“

Step 6: Now create the table structure:

In the SQL Query Analyzer open the script *create.sql* in the LPWAS directory (e.g. *c:\inetpub\wwwroot\lpweb\db\create.sql*)

Step 7: Execute it by using the menu item „*Query→Execute*“ (green arrow)

Important: The target database has to be selected correctly (on the right side of the green arrow)



Now the database setup is complete. Continue with the customization of the config-files.

Note
If you do not want to use the user sa for this database then create a user who has full read and write access.

2.2.5 Using a Hosting Service

If you use an external hosting service for the Web Activation Service the installation is a little bit different.

Note
Mirage offers a complete hosting service where you just have to deliver the configuration files. The complete installation including the setup of the database is done by Mirage.

Perform the following steps to use a hosting service:

- Install the MSI file on your local machine (or send the MSI file to your provider for installation)
- Copy the complete folder structure of your local Web directory with the LPWAS to your provider
- Set up the database as described

Your **provider** has to manually add the application to the IIS-server as follows

To create an application in IIS you have to perform the following steps (for example lpweb, installed in c:\inetpub\wwwroot\lpweb):

For Windows 2000 and XP Pro:
- Start "Internet Information Services" (Start - Settings - Control Panel - Administrative Tools)
- Navigate to the MyApp virtual directory and select it
- Right click it and select "Properties"
- Press the "Create" button and accept the default name
- Press OK

For NT4 Server:
- Start "Internet Service Manager" (Programs - Administrative Tools)
- Navigate to the MyApp virtual directory
- Right click it, and select "Properties"
- Press the "Create" button and enter a name, for example "MyApp"
- Press OK

If you use a hosting company and don't know how to do this you can contact their helpdesk about it. If they tell you they can't because it would influence other sites: not true, because that obviously does not depend on the presence of an application. Besides, each site can be configured to run in its own isolated memory space in IIS.

The next step is that your **provider** has to set the proper rights for LPWAS.

The ASP application works with an IIS-process 'user'. This user must have full read and write. The user name is **IWAM_Machine Name**. These rights should be given to the complete lpweb folder with all subfolders. The same is for the user **IUSR_Machine name** and user **ASP.NET** access.

- Choose the lpweb folder
- Right-click and choose Properties
- On the "Directory Security" tab you can edit the access rights
- Set all rights to full access
- Click Advanced
- Select Inherit from parent the permission entries that apply to child objects

2.2.6 Customize the configuration files

The following configuration files have to be configured

- The `web.config` file
- The `licprotectorws.config` file
- The `log.config` file
- One or more Project files
- One or more Automatic Licence Generator configuration files

Configuring web.config

This file resides in the root directory of the web application (e.g. c:\inetpub\wwwroot\lpweb\web.config)

Here the following important settings are done:

1. Define where the log-configuration can be found: Simply adapt the value of the key „logconfig“ to your Web Application path (e.g. c:\\inetpub\\wwwroot\\lpweb). Note to use two backslashes in the path-value

2. Define where the Licence Protector LPWAS configuration can be found: Simply adapt the value of the key „licprotectorwsconfig“ to your Web Application path (e.g. c:\\inetpub\\wwwroot\\lpweb). Note to use two backslashes in the path-value

3. The database connection string: Set the value of the key „db“ according to your database position.
   Set „user id“ to a user who has **read and write access** to your LPWAS database. The system administrator „sa“ is not a bad choice.
   Set „password“ to the password of the configured user.
   Set „data source“ to your SQL Server machine.
   Set „initial catalog“ to your LPWAS database-name. Normally that name is LPWeb23.

Here is the configuration for our example. The Web Application is installed in c:\inetpub\wwwroot\lpweb, the database machine is „mysqlserver“, the database is named „LPWeb23“ and the user „sa“ has the password „secret“:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <appSettings>
        <add key="logconfig" value="c:\\inetpub\wwwroot\\lpweb\\log.config" />
        <add key="licprotectorwsconfig"
            value="c:\\inetpub\wwwroot\\lpweb\\licprotectorws.config" />
        <add key="db" value="user id=sa;data source=mysqlserver;persist
            security info=False;initial catalog=LPWeb23;password=secret" />
    </appSettings>
    <system.web>
…
```

You have at least to modify the settings for the database after the installation. It will not work with the default settings.

Configure projects

The configuration of the hosted projects has to be set up in file `licprotectorws.config`. This file can be found in the web application's root directory (e.g. c:\inetpub\wwwroot\lpweb). It is a XML-file with the following structure:

TAG	Level, [min,max]	Description	Comment
<Configuration> *	0 [1,1]	Surrounds the whole configuration	
<Project> *	1 [1,n]	Surrounds a project definitions.	
<ProjectID>	2 [1,1]	Name or identifier of the project. This value is used to identify a project and therefore should be unique.	The Licence Protector calls to the Web Activation always have to include this Project ID (it is called ProjectName inside Licence Protector) Example: Demo project
<CheckActivationKeys>	2 [0,1]	A yes/no value that determines if the keys are checked for that project. It's the main switch for checking Activation Keys.	Default: true
<NoOfAllowedKeyOccur>	2 [0,1]	Number of times the same key is allowed to be checked. To ensure that a key can only be applied one time this value should be set to 1.	Default: 1

<CheckEntries>	2 [0,1]	A yes/no value that determines if the entries are checked for that project (WebRegister-Call). It's the main switch for WebRegister.	Default: true
<EntryRequired>	2 [0,1]	A yes/no value that determines if the sent ID has to exist in the tblEntry-Table. If the serial number functionality should be used this value has to be set on.	Default: false
<NoOfAllowedEntryOccur>	2 [0,1]	Number of times the same ID is allowed to be activated.	Default: 1
<InstCodeType1 .. 3>	2 [0,1]	The priority of used installation code types. The first installation code that is *not empty* is used for Copy Protection. **If you want to create keys without copy protection do not specify this entry.**	Example: InstCodeType1 is 2 (MAC-Address) InstCodeType2 is 1 (Volumenumber) InstCodeType3 is 3 (Hostname) If in a request InstallationCode 2 (MAC) is not empty, it is used for creating Copy protection keys. If it is empty (like in Windows 98 or Windows NT clients) then the next type is used: At our example this is type 1.
<GeneratorPath>	2 [0,1]	Path (without application) to the Automatic Licence Generator that is used to produce keys for that project.	Example: c:\inetpub\wwwroot\bin\alg
<GeneratorApp>	2 [0,1]	Filename of the Automatic Licence Generator that is used to produce keys for that project. This binary has to be found in the GeneratorPath.	Example: AutoLicGenerator.exe
<GeneratorConfig>	2 [0,1]	Filename of the configuration-file for the Automatic Licence Generator that is used to produce keys for that project. This file has to be found in the GeneratorPath.	Example: config-autogenerator.xml

Note:
The GeneratorXXX TAGs allows each project to use different configurations and different versions of AutoLicGenerator. Your application (project A) could run with version 2.2, your application B with version 2.3.

Here is our example file licprotectorws.config:

```
<?xml version="1.0" encoding="utf-8" ?>

<Configuration>
   <Project>
      <ProjectID>Demo Project</ProjectID>

      <CheckActivationKeys>yes</CheckActivationKeys>
      <NoOfAllowedKeyOccur>1</NoOfAllowedKeyOccur>

      <CheckEntries>yes</CheckEntries>
      <EntryRequired>no</EntryRequired>
      <NoOfAllowedEntryOccur>1</NoOfAllowedEntryOccur>

      <InstCodeType1>2</InstCodeType1>
      <InstCodeType2>1</InstCodeType2>
      <InstCodeType3>3</InstCodeType3>

      <GeneratorPath>C:\inetpub\wwwroot\lpweb\bin\alg</GeneratorPath>
      <GeneratorApp>autolicgenerator.exe</GeneratorApp>
      <GeneratorConfig>config-autogenerator.xml</GeneratorConfig>

   </Project>
</Configuration>
```

Configure logging

If you do not change the delivered file log.config the following default values are set:

- Your Logfile is named lpws.log and resides in the applications root path (e.g. c:\inetpub\wwwroot)
- The log level is at highest so you will get a large log file after some time

In your web.config file in key „`logconfig`“ you have configured the configuration file of the logging mechanism. Typically this file is named „`log.config`“. Now you can adapt this configuration:

1. Determine the name of the created logfile in the value of parameter „`File`“. If you use a name without path information the log file is written in your Web Application's root path (e.g. *c:\inetpub\wwwroot\lpweb*). Default is „`lpws.log`“.
2. Determine the level of the logged information. The highest level is „*DEBUG*“. This setting should be used if you experience problems and want to contact our support.
   If the system runs stable you can set this level down to „*INFO*“, „*WARN*“ or even „*ERROR*“.
   At those settings you will get much less lines of logged information in your logfile.

Here is the delivered log configuration file:

```
<log4net>
    <appender name="FileAppender" type="log4net.Appender.FileAppender">
        <param name="File" value="lpws.log" />
        <param name="AppendToFile" value="true" />
        <layout type="log4net.Layout.PatternLayout">
            <param name="ConversionPattern" value="%d [%t] %-5p %c [%x] - %m%n" />
        </layout>
    </appender>
    <root>
        <level value="DEBUG" />
        <appender-ref ref="FileAppender" />
    </root>
    <!-- Print only messages of level INFO or above in the package de.subware.common.
    swbase -->
    <logger name="de.subware.common.swbase">
        <level value="INFO" />
    </logger>
</log4net>
```

Configure the Automatic Licence Generator

With the file *licprotectorws.config* you have configured your project(s). You have set up a path <GeneratorPath> (e.g. c:\inetpub\wwwroot\lpweb\bin\alg). In that path LPWAS expects an Automatic Licence Generator application (ALG) with all necessary configuration files. These are at least a *config-autogenerator.xml* and a project configuration (e.g. Demo.xml) file.

The complete description how to set up and configure the files can be found in the *Licence Protector – Automatic Licence Generator Documentation.*.

Note:

- You should configure the ALG to generate Activation Keys. **Generated licence files are not used by LPWAS.**

- **The ProductID in the config-autogenerator.xml has to match the Module ID of the module that has to be activated**

- After you have configured the necessary files you should check and test your system.

2.2.7 Check the System set up

LPWAS has an included check mechanism. It is called TestConfiguration and can be reached with a web browser on the server machine.

Open the browser and type http://localhost/lpweb/lpws.asmx . You will see a list of all calls LPWAS provide. Click on **TestConfiguration**.

Now you can enter two parameters: The project and the ModuleID. In the provided Demo Application you should enter “Demo Project” (case sensitive!) and “D1001”. Press the button to start the test.

A WebRegister call for the module in the project is now simulated. You see the output in a new browser window:

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://www.mirage-systems.de/">Checking log.config: configured file
  'C:\\Inetpub\\wwwroot\\lpweb\\log.config' found. Checking licprotectorws.config: configured file
  'C:\\Inetpub\\wwwroot\\lpweb\\licprotectorws.config' found. Checking database: Reading...ok Writing...ok
  Testing project 'Demo Project': Test WebRegister for project Demo Project and module D1001: successful: Set
  WebActivation to 2. Created 2 keys: Key 1: 8CDsW-ICq93-jG6IS-4CMOt-zJDii-6J1k6 Key 2: I7D2I-5CLj3-9G5I7-
  3CBO7-fJD5i-6K14G No problem recognized at testing configuration. Continue with further setup tests.</string>
```

If the string ends with “Continue with further setup tests.” then the System seems to be set up correctly. Nevertheless you should do further tests. Otherwise have a look at the log file.

Note:

- Check if the system has **write access** to the LPWAS’s directories: After the TestConfiguration you should be able to find the log-file lpws.log in the application’s root directory. The system has to write the order file for the Automatic Licence Generator in the GeneratorPath (e.g. c:\inetpub\wwwroot\lpweb\bin\alg)
- If you get a database error check if the user has read and write access and if all parameters, including the name of the database server, are correct
- Do you have provided a valid LicProtector.lic file in the GeneratorPath for the AutoLicGenerator?
- Check the log file lpws.log for errors.

3 System Overview

3.1 The Data Model

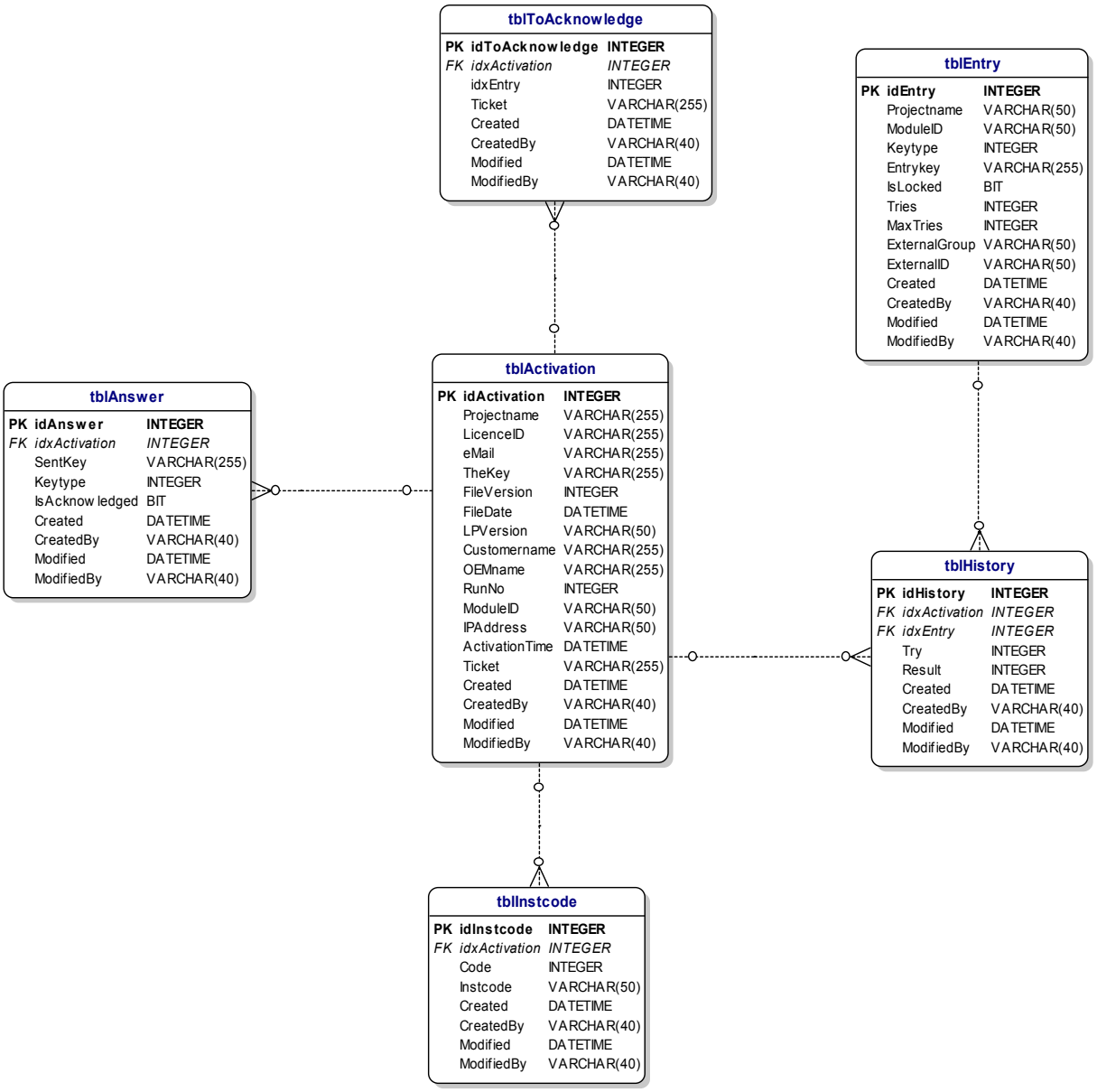


Table	Purpose	What can I do with this table
tblActivation	Every Request to LPWAS is logged this table	Reports who has tried to activate a module or check a key.
tblInstcode	Every Instcode belonging to a client that makes a request (which is protocolled in	You see what Installation Codes your customers have.

	tblAvtivation) is protocolled here	
tblEntry	All IDs (e.g. serial numbers) that are already used or are allowed to be used	Check if a user is already registered. Check how many tries are left. Check if an entry is locked. Increase/decrease remaining activations.
tblHistory	Links an entry (from tblEntry) to an Activation in tblActivation	You see what Activation leads to an entry.
tblAnswer	Protocolls all keys that are generated and send	Who has received which keys by LPWAS
tblToAcknowledge	Necessary for an Undo of an Activation if there are failures in applying keys	All records in that table are open or failed requests. To clean up use function `Refresh`

3.2 How does the Web Activation Server work?

Only 1 DLL call is needed to communication with the Web Activation Server – ***WebRegister***. All other necessary information like the http address of the Web Activation Server can be stored within the licence file.

3.2.1 Activate a module with WebRegister

Your application has to use the function ***WebRegister*** to make a request to the Web Activation Server.The important parameters are: **Projectname**, **ModuleID** and **Entrykey.**

Projectname

The Projectname is used to determine the project configuration in *licprotectorws.config*. The project name should be identical with the project name of your template xml file but it can be different.

Example: You want to implement a different workflow for the activation of the full and the demo version of a product. Therefore you would have to define 2 projects on the Web Activation Server but you would still have 1 project file for generating the licence file.

You can add any number of projects you like. It is not counted as a licence like it is within the Licence Generator.

ModuleID

Activation is done on module level. Therefore the module number = moduleID is necessary. If you just want to activate (e.g. set copy protection) a licence file, then you should use your main module for the activation.

Entrykey

The EntryKey is a **unique identifier**. The ID is used to identify the customer. Normally the unique ID is the licence file ID (you can obtain the ID with **GetLicFileID**). If you do not have a serialised licence file, then the unique ID could be the e-mail address, order number or the installation code (e.g. use **GetInstcode** and combine all installation codes to 1 string).

The LPWAS checks if this entrykey is **allowed to be activated**. In tblEntry a record is searched for the **combination Projectname, ModuleID and entrykey**. If no record is found it means that this entrykey is checked the first time. A new record is created. If a record is found then it is checked if it is **locked** or the **number of trials is greater than the default number of trials** (set in project configuration licprotectorws.config) or greater than an **individual limit**. If the limit is not exceeded and the record is not locked, the entrykey is inserted and the activation action is free to take place. Otherwise the call returns an error.

Automatic Licence Generator (ALG)

The activation action is invoking the *Automatic Licence Generator* (..\alg\autolicgenerator.exe). This is the universal batch processing tool to generate a licence file or key. It can be obtained as a separate module. In the combination with the Web Activation Server it can only be used with the server and not as a standalone tool. The standalone tool requires a licence for the Automatic Licence Generator.

The ALG processes the configuration.xml file defined for this project (within licprotectorws.config parameter <GeneratorConfig<GeneratorConfig>) and looks for the definitions defined for the **module ID**. Then the project template file is processed.

The complete description how to configure the config-autogenerator.xml project file is described within the documentation *Licence Protector – Automatic Licence Generator.pdf*.

The generated keys are sent to the calling application via web and applied on the licence file. The calling Licence Protector acknowledges the appliance. If a key could not be applied then the try is removed so that this entry could be used for another (successfully) try.

Registration of pre-registered users

If you want to allow registration only for pre-registered users/values (an order number or a key for identification can only be used if it is already stored in the Web Activation Server) you have to upload these values on the server. Note: The EntryKey is the unique ID for this value.

If in the project configuration the TAG *EntryRequired* is set to true, then the system behaves different. If a search for the combination Projectname, ModuleID, **entrykey finds nothing** then the call will return an error.

In that mode all possible entrykeys for a module in a project have to be in the **table tblEntry** before the WebRegister call for that entrykey is done. This workflow allows to load a set of valid serial

numbers in tblEntry and only **existing ID's that are never used before** will be allowed to be activated.

Samples for Activation of a Module

Here is the sample used in our Demo LPWAS service on http://kane1.dynalias.net/lpweb/lpws.asmx: See document ***Licence Protector – Web Activation Step by Step.pdf*** to test this sample with the licence-tester.

The server sample files are in the directory \Inetpub\wwwroot\lpweb\bin\alg

For every module that has to be activated, a product configuration is made.

File config-autogeneratorDemo.xml

```
<Connector>

   <Product>
      <ProductID>D1001</ProductID>
      <ProductName>LicenceProtector WebRegister Demo</ProductName>

      <ProjectFilename>demo.xml</ProjectFilename>
      <LicenceFileID>no</LicenceFileID>

      <LicenceFile>no</LicenceFile>          'Licence File generation can not be used in LPWAS
      <ActivationKey>yes</ActivationKey>     'Activation Keys must be turned on

      <CopyProtection>1</CopyProtection>     'A default Copy Protection type is set. The used value will be provided from the project configuration.
      <CreateCopyProtectionKey>yes</CreateCopyProtectionKey>  'Copy Protection is activated for the whole licence file. Note that the Copy Protection feature is not bound to one module (here D1001) but to the licence file. The activation of this module is used to activate the copy protection.

      <Modules>
         <Module>
            <ModuleNumber>D1001</ModuleNumber>  'the module configuration for the key. In a LPWAS config-autogenerator.xml the ModuleNumber typically is equal to the ProductID
            <Demoversion>No</Demoversion>     'Demoversion is set off. Note that all time limitations are removed when such a key is applied. If you do not want this then omit the value <Demoversion>
            <Value>1</Value>          'One licence is added
            <Days>10</Days>           'A 10 day period is provided
         </Module>
      </Modules>
   </Product>
```

```
<Product>
   <ProductID>D1002</ProductID>
   <ProductName>LicenceProtector WebRegister Demo</ProductName>

   <ProjectFilename>demo.xml</ProjectFilename>
   <LicenceFileID>no</LicenceFileID>

   <LicenceFile>no</LicenceFile>
   <ActivationKey>yes</ActivationKey>

   <Modules>
      <Module>
         <ModuleNumber>D1002</ModuleNumber>
         <Demoversion>No</Demoversion>
         <Value>5</Value>
         <FixedValue>Yes</FixedValue>
         <Days>10</Days>
      </Module>
   </Modules>
</Product>
<Product>
   <ProductID>D1003</ProductID>
   <ProductName>LicenceProtector WebRegister Demo</ProductName>

   <ProjectFilename>demo.xml</ProjectFilename>
   <LicenceFileID>no</LicenceFileID>

   <LicenceFile>no</LicenceFile>
   <ActivationKey>yes</ActivationKey>

   <CopyProtection>0</CopyProtection>
   <CopyProtectionX>3</CopyProtectionX>
   <CreateCopyProtectionKeyX>yes</CreateCopyProtectionKeyX>

   <Modules>
      <Module>
         <ModuleNumber>D1003</ModuleNumber>
         <Demoversion>No</Demoversion>
         <Value>3</Value>
         <FixedValue>No</FixedValue>
         <Days>13</Days>
      </Module>
   </Modules>
</Product>
```

```
   <Product>
      <ProductID>D1004</ProductID>
      <ProductName>LicenceProtector WebRegister Demo</ProductName>

      <ProjectFilename>demo.xml</ProjectFilename>
      <LicenceFileID>no</LicenceFileID>

      <LicenceFile>no</LicenceFile>
      <ActivationKey>yes</ActivationKey>

      <CopyProtection>0</CopyProtection>
      <CopyProtectionX>3</CopyProtectionX>
      <CreateCopyProtectionKeyX>yes</CreateCopyProtectionKeyX>

      <Modules>
         <Module>
            <ModuleNumber>D1004</ModuleNumber>
            <Demoversion>No</Demoversion>
            <Value>3</Value>
            <FixedValue>No</FixedValue>
            <Days>13</Days>
         </Module>
      </Modules>
   </Product>
</Connector>
```

File Demo.xml

The values in blue are important values for the Web Activation. They can be set vial DLL functions but we recommend to set it within the project file. The settings are then stored within the licence file during the generation of the file.

```
<LicenceProtector>
       <Default>
              <ProjectName>Demo Project</ProjectName>
              <ProjectSecureKey>What a beautiful morning</ProjectSecureKey>
              <NameLicenceFile>MyApplication</NameLicenceFile>
              <TAGValue>This is my TAG</TAGValue>
              <CopyProtection>0</CopyProtection>
              <ListBoxManufacturer>Mirage Computer Systems GmbH</ListBoxManufacturer>
              <CustomerName>Trial Version</CustomerName>
              <DefaultDiffPct>0</DefaultDiffPct>
              <WebServiceURL>http://kane1.dynalias.net/lpweb/lpws.asmx</WebServiceURL>
              <ShowWAStartPage>yes</ShowWAStartPage>
              <ShowWAProgressPage>yes</ShowWAProgressPage>
              <ShowWAResultPage>yes</ShowWAResultPage>
```

```
</Default>
<LicenceModules>
        <Module>
                <ModuleID>D1001</ModuleID>
                <ModuleName>ERP Module</ModuleName>
                <ModuleType>1</ModuleType>
                <Value>5</Value>
                <AbsoluteLics>no</AbsoluteLics>
                <Days>30</Days>
                <AbsoluteDays>no</AbsoluteDays>
                <ValidUntilDay>31.12.2005</ValidUntilDay>
                <TAGValueModule>10</TAGValueModule>
                <Demoversion>yes</Demoversion>
                <LoadAsDefault>Yes</LoadAsDefault>
                <AllowDeactivate>yes</AllowDeactivate>
        </Module>
        <Module>
                <ModuleNumber>D1002</ModuleNumber>
                <ModuleName>Analysis Module</ModuleName>
                <ModuleType>2</ModuleType>
                <Value>2</Value>
                <Demoversion>yes</Demoversion>
                <TAGValueModule>20</TAGValueModule>
                <LoadAsDefault>Yes</LoadAsDefault>
                <AllowDeactivate>yes</AllowDeactivate>
        </Module>
        <Module>
                <ModuleNumber>D1006</ModuleNumber>
                <ModuleName>Accounting Module</ModuleName>
                <ModuleType>6</ModuleType>
                <Value>2</Value>
                <Demoversion>yes</Demoversion>
                <TAGValueModule>20</TAGValueModule>
                <LoadAsDefault>Yes</LoadAsDefault>
        </Module>
        <Module>
                <ModuleNumber>D1003</ModuleNumber>
                <ModuleName>Number of Employees</ModuleName>
                <ModuleType>3</ModuleType>
                <Value>100</Value>
                <TAGValueModule>30</TAGValueModule>
                <LoadAsDefault>Yes</LoadAsDefault>
        </Module>
        <Module>
```

```
                <ModuleNumber>D1004</ModuleNumber>
                <ModuleName>CTI Module</ModuleName>
                <ModuleType>4</ModuleType>
                <Value>YES</Value>
                <TAGValueModule>40</TAGValueModule>
                <LoadAsDefault>yes</LoadAsDefault>
        </Module>
        <Module>
                <ModuleNumber>D1005</ModuleNumber>
                <AllowDeactivate>yes</AllowDeactivate>
                <ModuleName>Custom Reports</ModuleName>
                <ModuleType>5</ModuleType>
                <Value>3</Value>
                <TAGValueModule>50</TAGValueModule>
                <LoadAsDefault>yes</LoadAsDefault>
        </Module>
    </LicenceModules>
</LicenceProtector>
```

3.2.2 Online Validation of an Activation Key

The information whether a key is verified online or not is **stored within the key**. The parameter is set when the Activation Key is generated. Your application therefore has to do 'nothing special' if the projectname is identical to the project name used during creating of the licence file and the Web Activation Server http address is already stored within the licence file (recommended).

If the customer applies the key via the Licence Viewer, then the Licence Viewer handles everything. If the key is applied with the function *ApplyActivationKey* or *ApplyActivationKeyForProject* then the DLL function handles everything.

With that function call no activation action exists. It is only checked if the number of times this key was already used is less then the configured number of maximum allowed usages (normally =1). If the key usage is within the defined limit, the key is stored in the database and the invoking Licence Protector application will be allowed to apply that key. Otherwise this call will return an error and the key can not be applied.

The definition for applying a key is done within the licprotectorws.config file

```
<Configuration>
  <Project>
    <ProjectID>Demo Project</ProjectID>
```

```
    <CheckActivationKeys>yes</CheckActivationKeys>
    <NoOfAllowedKeyOccur>1</NoOfAllowedKeyOccur>

    <CheckEntries>yes</CheckEntries>
    <EntryRequired>no</EntryRequired>
    <NoOfAllowedEntryOccur>1</NoOfAllowedEntryOccur>

    <InstCodeType1>2</InstCodeType1>
    <InstCodeType2>1</InstCodeType2>
    <InstCodeType3>3</InstCodeType3>

    <GeneratorPath>C:\Inetpub\wwwroot\lpweb\bin\alg</GeneratorPath>
    <GeneratorApp>autolicgenerator.exe</GeneratorApp>
    <GeneratorConfig>config-autogeneratorDemo.xml</GeneratorConfig>

  </Project>

</Configuration>
```

4 Integrate LPWAS Features in your application

A complete function reference is listed in the main ***Licence Protector Developer documentation***.

4.1 Set connection parameters

If you want to use LPWAS in your application you have to configure the invoking Licence Protector application. Therese configurations can be done via DLL functions or as default settings within the licence file.

There are the following important settings:

- The URL where your LPWAS service can be found
- The Proxy settings that enable the contact to this LPWAS service
- The Projectname which determines what project is used on the LPWAS service

If you turn off the connection form by setting *ShowWAStartPage* to false then you have to provide these connection parameters manually.

4.1.1 URL to the LPWAS service

Licence Protector has a property *WebServiceURL*. To reach your LPWAS service you have to provide the URL where this service can be found in the internet. This URL can be stored inside the licence file.

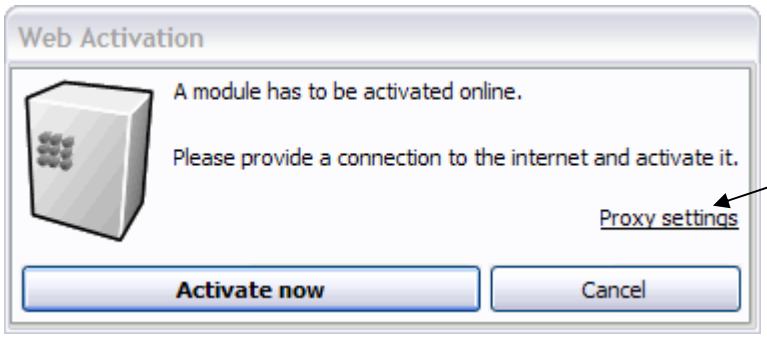
There are 2 possibilities to set this URL:

1. The WebServiceURL is set at generating the licence-file. It is provided with the TAG <WebServiceURL>. See also the documentation of project file and configuration file/order file of Automatic Licence Generator

2. The WebServiceURL is set by the application using the property *WebServiceURL*. Note that after setting this property the URL is stored inside the licence-file. Be aware that you should not use different URLs for clients that share one licence file

4.1.2 Proxy settings

If you use the standard dialog (see below) the user can configure the proxy settings within this dialog. The default settings from the Browser are used. Your application does not need to configure any settings via DLL functions.



The following properties can be used to control proxy settings via DLL functions:

- UseProxyServer : Main switch to determine if a proxy should be used
- UseIEProxyServer : If UseProxyServer is on then this value determines if the proxy settings from Microsoft Internet Explorer should be used as LPs proxy settings
- ProxyServer : If UseProxyServer is on but UseIEProxyServer is off then this value determines the IP or name of the proxy server that should be used to access the LPWAS service
- ProxyPort : If UseProxyServer is on but UseIEProxyServer is off then this value determines the port number of the proxy server set with ProxyServer.

All these properties are stored locally in the registry. So every user or client can have its own settings.

Normally the LPWAS service can be reached from all machines that can access the internet. If a client is connected directly to the internet you do not need to use a proxy server. In that case turn UseProxyServer off. The other settings are meaningless in that case.

If a client can reach the internet only via a proxy server, you can turn on UseIEProxyServer, and then the configured proxy information about proxy server and port are used to reach the LPWAS service. ProxyServer and ProxyPort are ignored if UseIEProxyServer is on.

If the Microsoft Internet Explorer is not configured or you want to explicitly control which proxy server is used to reach the LPWAS service, then turn UseIEProxyServer off and set the values to ProxyServer and ProxyPort.

4.1.3 Projectname

LPWAS can have several projects which behave different. To control which project on the LPWAS server should be used you have two possibilities:

- If you set an empty value in the parameter projectname then **the internal projectname of the licence file is used** (recommend). Normally that name should be set at generating the licence file. The TAG <Projectname> is used to set this value. See main Licence Protector documentation
- Explicitly set the projectname in the calls WebRegister and ApplyActivationKeyForProject.

You can set the projectname with the property Projectname. Note that this will change the value inside the licence file.

If the internal projectname is not set, reading the property Projectname will return the name of the licence file without ending. For example: An empty Projectname in licence file MyApp.lic will return the Projectname MyApp.

We strongly recommend setting the projectname at generating the licence-files.

4.2 Determine, if a module has to be activated: GetWebActivation

To check if a module has to be activated use *GetWebActivation*(module number).

A module needs to be activated if the return value is 1 (required) or 3 (erroneous – problem occurred during a previous web activation).

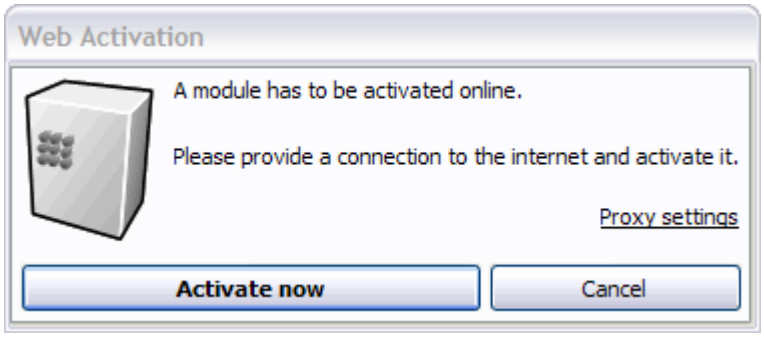
Note
The web activation state is only status information what to do. It triggers no automatic action. If a module has state 1 there is no implicit WebRegister call.

4.3 Activate a module: WebRegister

The call *WebRegister* requests Activation for the project and the module. An entrykey has to be set to check it with table tblEntry.

The actions in call WebRegister(“”, “D1001”, “1234567890”, “”) are as follows:

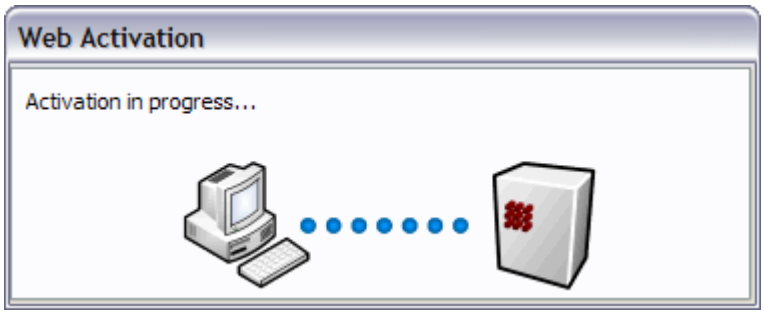
- First a connection window appears unless you have turned it off by setting *ShowWAStartPage* to ShowWAResultPage to false. The user can configure proxy settings



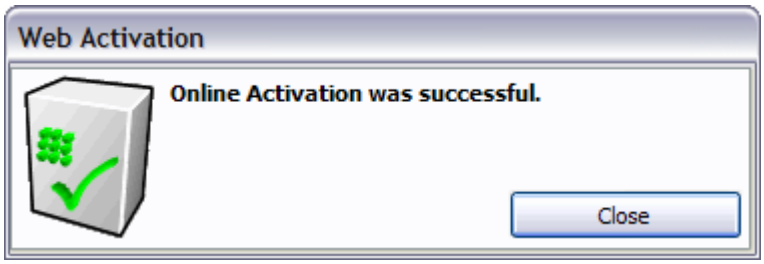
- Now the LPWAS is contacted. It is no explicit Projectname set, so the internal Projectname of the licence file is used. LPWAS reads the configuration for the project. If the Project could not be found the call will end with an error. In the next step the entry is checked in tblEntry. If it could be

found (in the project and modules section) then the number of tries is checked. If the entry is not locked and has at least a free try then the Automatic Licence Generator is called. The used product is the module delivered by the WebRegister call

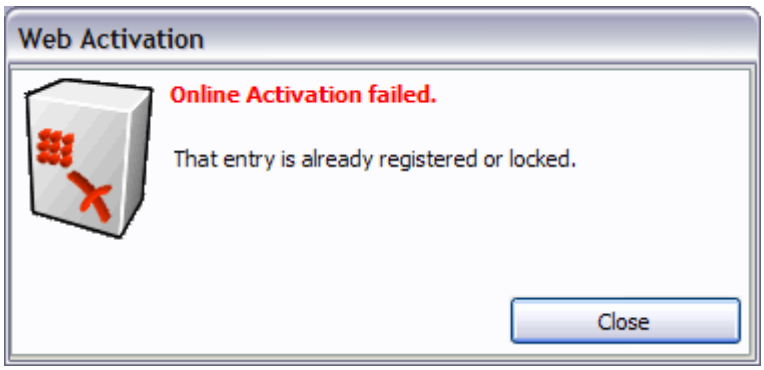
- The user sees an animated window that the Activation is in progress



- The generated Activation Key(s= are transferred to the local licence file and are tried to be applied. After that the WebRegister-call does an Acknowledge-Call to tell LPWAS that the output was successfully processed
- The WebActivation Status of the module is set to “done” (numerical 2)
- The user get an confirmation screen



If the Activation fails, the user get a screen with an error message (in the language defined with SetLanguage – default = English).



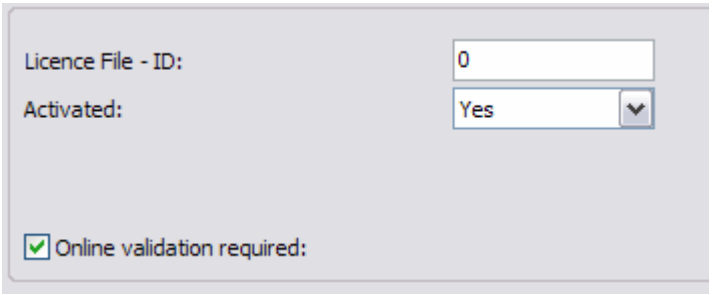
You can call WebRegister whenever you like but we recommend the following sequence:

- If you want a module to be activated set it's WebActivation state to "required" (1) (set WebActivation of all other modules to "none (0)). This is done while creating a licence file or with a key
- Check this status and if required do the activation by calling WebRegister. After a successful activation the state is changed to "done" (2). If the activation failed this state is switched to erroneous (3).

Note:
If you decide to activate a module by its WebActivation status don't forget to process state 3 (erroneous). Use it like state "required" (1) to trigger the WebRegister-call.

4.4 Applying Activation Keys

To validate an Activation Key online, you either have to activate this (checkbox Online validation required) within Licence Generator (see below)



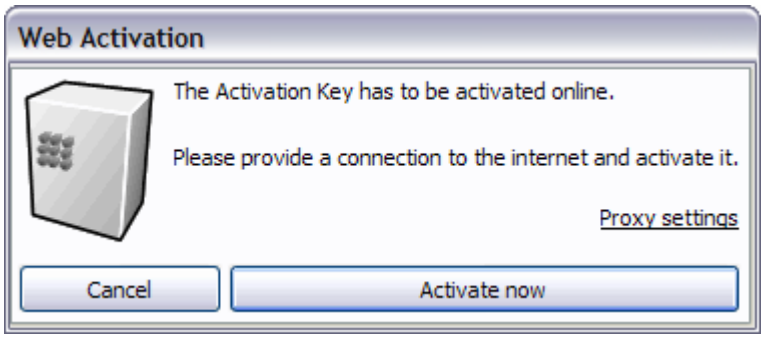
or set the TAG <ForceOnlineCheck> within the Automatic Licence Generator.

If you use such keys it is very important to have the internal projectname set correctly because functions that apply a key aren't able to explicitly set a projectname.

If the customer applies the key via the Licence Viewer, then the Licence Viewer handles everything. If the key is applied with the function *ApplyActivationKey* or *ApplyActivationKeyForProject* then the DLL function handles everything.

An exception is ApplyActivationKeyForProject in which the Projectname could be set as a parameter. This is required if the same programme has different project files on the LPWAS.

Identically to the *WebRegister* call the connection Window will be displayed if the Flags *ShowWAStartPage, ShowWAProgressPage* and *ShowWAResultPage* are set to on. The same applies for the proxy settings. On the text shown in the first window is slightly different.


Web Activation
The Activation Key has to be activated online.
Please provide a connection to the internet and activate it.
Proxy settings
Cancel
Activate now

5 Building a sample project

5.1 Planning…

Let's assume you have an application "STAR-App" that is protected by Licence Protector. Every customer gets a licence file with his own serial number (licence file ID) and you want to be sure that this licence file is only used once and not spread on the internet.

You have several modules and use Activation Keys to modify the number of licences. You also want to be sure that a Key is only used once.

With LPWAS you can accomplish this.

5.2 Configuring…

5.2.1 The Projectname

First you have to set a projectname: "STAR-App". That projectname has to be placed in your project configuration file **starapp.xml** in the TAG <ProjectName>.

The same projectname has to be used in the **licprotectorws.config** in the LPWAS root directory in the TAG <ProjectID>.

Note:
The ProjectName is case sensitive.

5.2.2 Configuration of licprotectorws.config

With the TAG <CheckActivationKeys> you configure that you want to validate keys online.

How often is the same key allowed to be applied? This is configured with the TAG <NoOfAllowedKeyOccur>.

To allow activating a module the TAG <CheckEntries> has to be set to yes.

How often is the same ID allowed to be activated? Configure it with the TAG < NoOfAllowedEntryOccur>.

If the ID has to be provided before activation with that ID can occur then turn this feature on in TAG <EntryRequired>

We want all to be checked, an Activation Key is once allowed to be applied, an Entry is allowed three times (we tolerate 2 alternative installations). See configuration file below…

licprotectorws.config

```
<?xml version="1.0" encoding="utf-8" ?>
<Configuration>
   <Project>
      <!-- This ProjectID has to be identical to the used ProjectName in the project file or in the Licence Protector calls -->
      <ProjectID>STAR-App</ProjectID>

      <!-- If Activation Keys should be checked then turn it on here -->
      <CheckActivationKeys>yes</CheckActivationKeys>
      <!—How often is the same Key allowed to be applied? Default is 1 for once. -->
      <NoOfAllowedKeyOccur>1</NoOfAllowedKeyOccur>

      <!-- If the Activation of modules should be checked then turn it on here -->
      <CheckEntries>yes</CheckEntries>
      <!-- Must an entry exist before it can be checked? This requires uploading the unique ID’s. -->
      <EntryRequired>no</EntryRequired>
      <!-- How often is the same ID allowed to be activated?  -->
      <NoOfAllowedEntryOccur>3</NoOfAllowedEntryOccur>

      <!-- Which Installation code type should be used 1st, 2nd …. -->
      <!-- If the first Installation Code is empty then the 2nd is used and so on. -->
      <!if no copy protection is required then omit this settings →
      <InstCodeType1>2</InstCodeType1>
      <InstCodeType2>1</InstCodeType2>
      <InstCodeType3>3</InstCodeType3>

      <GeneratorPath>c:\inetpub\wwwroot\lpweb\bin\alg</GeneratorPath>
      <GeneratorApp>autolicgenerator.exe</GeneratorApp>
      <GeneratorConfig>config-autogenerator.xml</GeneratorConfig>
   </Project>
   <Project>
      <ProjectID>Another project</ProjectID>
      …
```

5.2.3 Configure the Project XML file (template)

Beside the ProjectName it is necessary to configure the WebServiceURL and the layout of the Connection window. Within the project.xml file you can decide whether the start page (with the possibility to change the proxy-settings), the progress page and the result page should be displayed.

Our Main Module S001 is selected to be activated online. So we set the state of the WebActivation to 1 (Required)

If we generate a licence file for STAR-App all these settings are included in the lic-file. Nothing has to be configured using DLL calls during runtime.

Project file Star.xml

```
LicenceProtector>
   <Default>
      <ProjectName>STAR-App</ProjectName>
      <ProjectSecureKey>What a beautiful morning</ProjectSecureKey>
      <NameLicenceFile>starapp</NameLicenceFile>
      <ListBoxManufacturer>My Company</ListBoxManufacturer>
      <CustomerName>Trial Version</CustomerName>
      <WebServiceURL>
            http://www.mycompany.com/lpweb/lpws.asmx
      </WebServiceURL>
      <ShowWAStartPage>yes</ShowWAStartPage>
      <ShowWAProgressPage>yes</ShowWAProgressPage>
      <ShowWAResultPage>yes</ShowWAResultPage>
   </Default>
   <LicenceModules>
      <Module>
         <ModuleID>S001</ModuleID>
         <ModuleName>Main Module</ModuleName>
         <ModuleType>1</ModuleType>
         <Value>5</Value>
         <Demoversion>yes</Demoversion>
         <LoadAsDefault>Yes</LoadAsDefault>
         <AllowDeactivate>yes</AllowDeactivate>
         <WebActivation>required</WebActivation>
      </Module>
      <Module>
         <ModuleNumber>S002</ModuleNumber>
         <ModuleName>Report Module</ModuleName>
         <ModuleType>5</ModuleType>
         <Value>10</Value>
         <LoadAsDefault>Yes</LoadAsDefault>
         <AllowDeactivate>no</AllowDeactivate>
      </Module>
   <LicenceModules>
<LicenceProtector>
```

5.2.4 Choosing the ID to activate…

Decide what you want to use as a unique identifier for activating a module. You're free to decide whether you want to us an eMail-Address, an Invoice-Number, the Licence-File-ID, a serial number, even an Activation Key.

Serial numbers can be used with two flavours

- If your serial numbers have integrated checks, so that no one can "invent" a new one you can use the LPWAS feature with <EntryRequired> set to false: If a serial number is new, it is inserted in the entry-table. If it comes again it will be blocked.
- If your serial numbers do not have the possibility to check them then it is possible to simply use a new one. In that case you need a list of valid serial numbers before the Customers start to activate modules. An faked serial number is blocked because it does not exist.

The easies guess is to use the serial number of the licence file (licence file ID). You can retrieve it with *GetLicFileID.*

5.2.5 Activating module S001 means…

If the used ID is accepted, LPWAS will generate some keys for the calling licence-file. This is done by using the Automatic Licence Generator. Therefore you have to provide a configuration file **config-autogenerator.xml**. The product(s) in this file represent the Module ID for which this Activation call is made. You have to define which workflow (creating keys) has to be done when the module is activated.

In our sample, activating module S001 deletes the Demoversion entry (set to No - this also deletes time limitation) and the software copy protection is activated.

Config-Autogenerator.xml

```
<Connector>
   <Product>
      <ProductID>S001</ProductID> must match the module number
      <ProductName>STAR-Application – Basic Version</ProductName>
      <ProjectFilename>star.xml</ProjectFilename>
      <LicenceFileID>no</LicenceFileID>

      <LicenceFile>no</LicenceFile>
      <ActivationKey>yes</ActivationKey>

      <CopyProtection>2</CopyProtection>
      <CreateCopyProtectionKey>yes</CreateCopyProtectionKey>

      <Modules>
```

```
        <Module>
            <ModuleNumber>S001</ModuleNumber>
            <Demoversion>No</Demoversion>
        </Module>
    </Modules>
</Product>

</Connector>
```

5.2.6 Important Configuration Links at a glance

The following table provides an overview about the most important configuration settings.

Item	From	is linked to Item	In
ProjectName	Project file → Licence file	ProjectID	Licprotectorws.config
Projectname	WebRegister-Call	ProjectID	Licprotectorws.config
Projectname	ApplyActivationKeyForProject	ProjectID	Licprotectorws.config
WebServiceURL	Project file → Licence file	LPWAS	
ModuleID	WebRegister-Call	ProductID	Config-autogenerator.xml
GeneratorPath	Licprotectorws.config	Directory where the ALG operation is done	
GeneratorApp	Licprotectorws.config	Filename of the used ALG	
GeneratorConfig	Licprotectorws.config	Config-autogenerator.xml	
Projectname + ModuleID + EntryKey	WebRegister-Call	Table tblEntry	Database
Projectname + EntryKey	ApplyKey with ForceOnlineCheck	Table tblEntry	Database

6 Tips, Ideas and How To

You find a lot of Tips, Ideas and How to in the Knowledge Base, chapter Web Activation.

6.1 Delete time limitation

If you want to delete the time limitation after the registration is done then set Demoversion=No.

6.2 Activate a module user, PC or concurrent user

If you activate or change a user, PC or concurrent user module (e.g. delete time limitation) then you have to set value=0 to prevent changing the number of users, PC's or concurrent users. If this value is not set in the definition file, then the value from the project file is used and added to the existing value.

```
<Connector>
  <Product>
    <ProductID>S001</ProductID>
    <ProductName>Star application – Basic Version</ProductName>
    <ProjectFilename>star.xml</ProjectFilename>
    <LicenceFileID>no</LicenceFileID>
    <LicenceFile>no</LicenceFile>
    <ActivationKey>yes</ActivationKey>
    <CopyProtection>2</CopyProtection> -> activate copy protection
    <CreateCopyProtectionKey>yes</CreateCopyProtectionKey>
    <Modules>
      <Module>
        <ModuleID>S001</ModuleID>
        <Demoversion>No</Demoversion> -> delete time limitation
        <Value>0</Value> -> does not change the number specified in the licence file
      </Module>
    </Modules>
  </Product>
</Connector>
```

6.3 How can I change the maximum number of registrations of a user?

The maximum number of registrations per PROJECT is defined in the main configuration file licprotectorws.config <NoOfAllowedEntryOccur>. This can be changed for a **single LICENCE FILE / MODULE** within the database. In the table tblEntry set MaxTries to the maximum number of allowed registrations for the selected licence file / module.

The same applies to Activation Keys. The maximum number in the main configuration is defined with <NoOfAllowedKeyOccur> and it can also be changed in the table tblEntry with the MaxTries value.

There will be a Web interface to change this value starting with version 2.3.3 of the Web Activation Server.

6.4 Web Activation - how to activate when no web access is available?

If a licence file requires web activation and no internet access is available, the licence can be activated as follows:

- Your customer transmits the installation code to you (e-mail, phone, fax). The installation code is displayed in the Licence Viewer
- Create a copy protection key within the Licence Generator
- Create an Activation Key which sets Demoversion=No (deletes time limitation) and changes the Web Activation status to DONE

6.5 Unlimited number of registrations - how it works

The number of allowed registrations is defined within the licprotectorws.config file with the parameter

<NoOfAllowedEntryOccur> (default = 1)

This limits the number of registrations. To turn this testing on, the parameter <CheckEntries> must be yes (default value).

If you want to allow an unlimited number of registrations per user, then just set <CheckEntries> to No. Activation Keys are handled the same way:

<NoOfAllowedKeyOccur> (default = 1)

If you want to allow to use an Activation Key unlimited times then set <CheckActivationKeys> to No.

6.6 Activate a Demo Version

You could allow activating a demo version and grant your customers more days for evaluation. Normally Demoversions do not have a unique licence ID. To prevent multiple activations, use as ID the combination of all 3 installation codes.

- Retrieve the installation code 1-3 with *GetInstcode*

- Combine all 3 codes to one code
- Use this code as ID with webregister

7 Troubleshooting

7.1 How to get support?

First have a look at Licence Protector's Knowledge Base in http://www.licence-protector.com, choose Support, Knowledge Base.

If you can't solve the problem make a support request - http://www.licence-protector.com, choose Support, Support Request. Provide a detailed description of the problem including all necessary configuration files and especially the log-file(s)

7.2 Error Codes

The LPWAS produces the following Returncodes

Errorcode	Description	Reason	Action
3001	Connection Error on client – LPWAS could not be reached	- Client could not reach the internet - Proxy-Settings are incorrect - No WebServiceURL set in licence file	-Enable a internet connection -Set WebServiceURL - Check if LPWAS is running
3002	Licprotectorws.config could not be read	- XML format incorrect	Check Documentation
3003	Insufficient project data in licprotectorws.config	- No ProjectID set	Check Documentation
3004	Unknown InstCode type in licprotectorws.config	- InstCode does not exist.	Check Developer Documentation
3005	No projects defined in licprotectorws.config	- No project could be found	Check Documentation
3006	Bad request data	- Projectname is empty - Reguest is tampered	Set Projectname before calling LPWAS
3007	Unknown Requesttype	- Call use an unknown requesttype	Are you using a new Licence Protector against an older version of LPWAS?
3008	Writing in database failed		Check setup Check availability of the database server
3009	Reading from database failed		Check setup Check availability of the database server
3010	Key already used – no further usage is allowed	- Entrykey is locked - Entrykey has exceeded the individual or default limit	React on that result by displaying a message

3011	Unknown project	- Project could not be found in licprotectorws.config	Check Documentation
3012	Bad output of AutoLicGenerator	- Generated output from AutoLicGenerator could not be read	Check setup, TestConfiguration
3013	AutoLicGenerator failed	- Configuration of AutoLicGenerator is incorrect	Check Auto Lic Generator documentation
3014	Security Error	- Incorrect call data	
3015	Key does not exist	- Expected Entrykey could not be found in tblEntry (“Serial Number”)	Display message Provide that EntryKey in tblEntry
3016	Output of AutoLicGenerator could not be read		Check setup
3017	No key output of AutoLicGenerator	- Incorrect product setting in config-autogenerator.xml	Check Auto Lic Generator Documentation
3998	Unknown error		Check log file
3999	Unhandled error		Check log file