LICENCE PROTECTOR

Automatic License Generator Documentation

LICENCE PROTECTOR Version: 5

Date: June 2016

This documentation and the accompanying material are for informational purpose only and property of Mirage Computer Systems GmbH, Aulendorf. Information in this document is subject to change without note. The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

Table of Contents

1 Introduction

This document describes how to set up the Automatic License Generator, the necessary configuration and the basic implementation of the product. It refers to other documentation which should be read in advance.

- For default shop Integration using a Serial Number Key see the Online Tutorial Chapter *Integration in Online Shops* http://helpserver.biz/onlinehelp/lp/easygo/5.0/help2000/integration_in_online_shops.html

1.1. LICENCE PROTECTOR Automatic License Generator

The Automatic License Generator allows creating licence files and Activation Keys in an automatic mode. It could be used for:

- Integration in e-shops. After a customer has bought a product, automatically a licence file or Activation Key is generated. There is a ready to go integration for the shops of **www.shareit.com** , **www.element5.com**, **www.cleverbridge.com**
- Automate updates, e.g. send all customer new licence files or activate a module for selected customers
- Integration in your ERP or CRM system. Create automatically a licence file or a key based on the customer data stored in your system
- Distribution via Internet. A customer types in his installation code on your Website and gets a licence file with copy protection

The Automatic License Generator can handle complex requirements, like:

- 1 product ordered – the licence file consist of several modules – e.g. the product *ERP Enterprise edition* consist of the modules ERP Module, Number of Employees, Custom Reports, Accounting Module
- 1 product ordered – the modules can have a different quantity – e.g. 1 product *ERP Enterprise edition* consist of the modules 1 ERP Module, 3 Number of Employees, 5 Custom Reports, 2 Accounting Module
- 2 products ordered – if they have the same **project file** then the generator creates one 1 licence file from this order – e.g. the products *ERP Enterprise Edition* and the product *Analysis Module* are ordered. Then all modules from both articles are mixed in 1 licence file
- Create a licence file and an Activation Key within one order – e.g. send the customer a basic licence file and activate multiple modules with keys
- 1 order consist of different products (**project file**s) – different licence files are created

This documentation has its focus on the Automatic License Generator. You need to know, how the main product – LICENCE PROTECTOR – works, which is described in detail in the document: *Licence Protector- Developer Documentation.pdf.*

1.1 Files to install

Automatic License Generator can be installed by copying the following files in **one directory**:

▶ Automatic License Generator	AutoLicGenerator4xx.exe
▶ licence file	alg.lic
▶ Product configuration	config-autogenerator.xml
▶ One or more Project configurations	aaa.xml, bbb.xml, …

You have to activate the license using a Serial Number. The Automatic Gnerator can be installed on a network drive. The LicProtector.DLL is not needed.

1.2 Configuration of the Automatic License Generator

Since Automatic License Generator is a **command-line tool**, you have to provide some parameters and configuration files in order to let the generator produce licence-files or Activation Keys.

There are three types of configuration files. All three are in XML-format:

- The order file / input file – any filename
- The **product configuration file** – **config-autogenerator.xml**
- The **project file** - any filename

We provide sample files for testing and use the sample files in this documentation.

Project files

- Demo.xml
- Textprocessor.xml

Order File

- Myorder.xml – Order file with samples for licence file und licence key generation
- Myorder-copy-protection-keyonly.xml – Order file with sample to create a copy protection key and an Activation Key
- Myorderkeylist.xml – File to produce a keylist with 100 keys

Configuration File

- Config-autogenerator.xml

Batch files

The batch files invokes the Automatic License Generator with an order file

- Process.bat – Batch file for using Myorder.xml file
- Process-copy-protection-keyonly.bat – Batch file using the Myorder-copy-protection-keyonly.xml file
- Process-keylist.bat – Batch file using the Myorderkeylist.xml file

With the file: **process.bat** the AutoLicenceGenerator.exe is started and processes the myorder.xml file. The following files are generated:

- Myapplication.lic -> licence file with modules from the demo.xml **project file**
- Myapplication_activation_key.txt -> File with Activation Key from the demo.xml **project file**
- Textproc.lic -> licence file licence file with modules from the textprocessor.xml **project file**

With the file: **process-copy-protection-keyonly.bat** the AutoLicenceGenerator.exe is started and processes the Myorder-copy-protection-keyonly.xml file. The following files are generated:

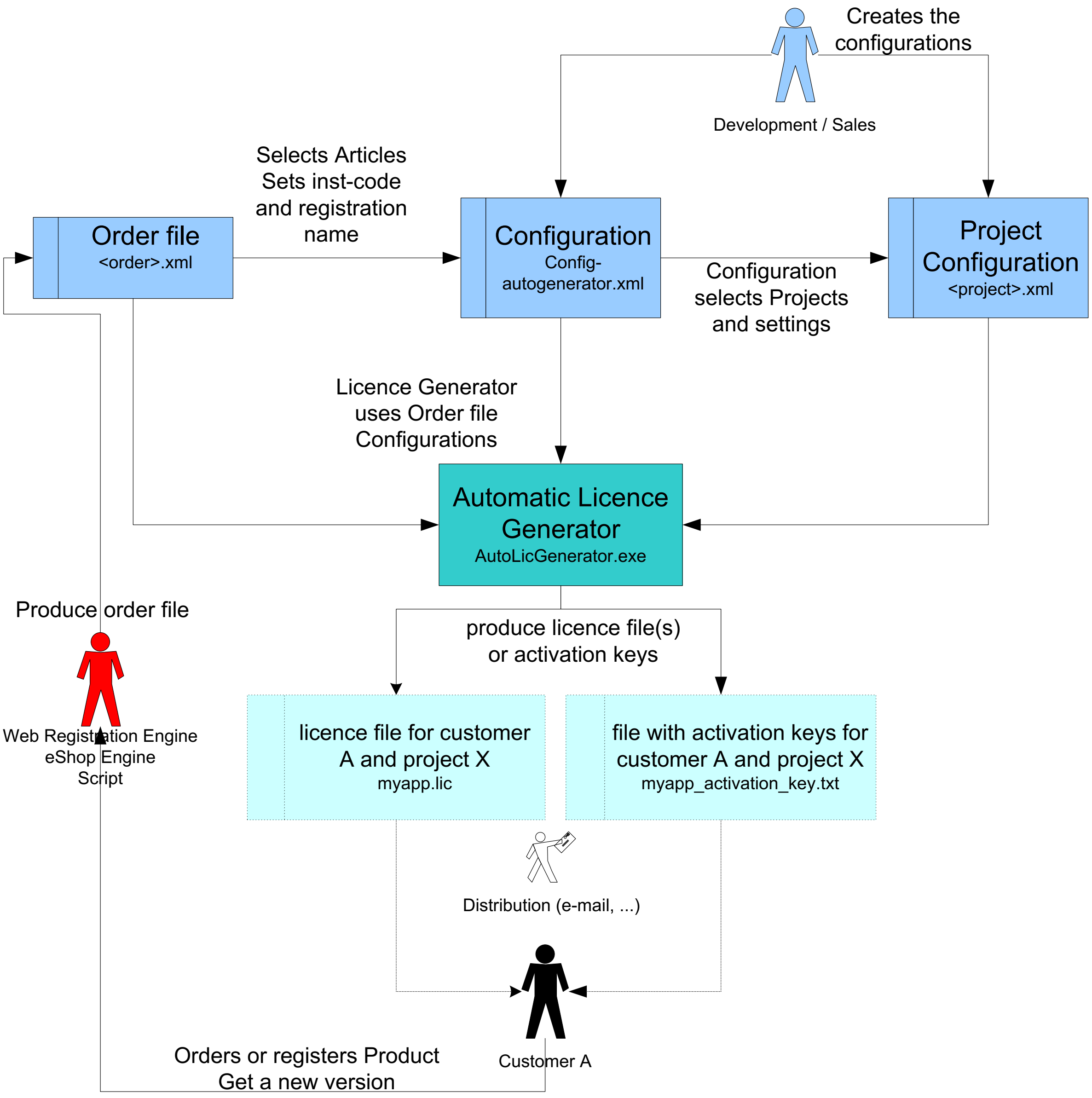
- Myapplication_activation_key.txt -> File with Activation Key for copy protection using installation code number 2 and Activation Key for CTI module

With the file: **process-keylist.bat** the AutoLicenceGenerator.exe is started and processes the Myorderkeylist.xml file. The file keylist.txt with 100 keys is generated.

Complete Website sample

The folder …\internet-sample contains a complete sample source how to invoke Licence Protector from an internet site and to produce a licence file or keys. The keys are sent automatically via the e-mail server or manually via the local e-mail application.

Process Overview


Creates the configurations
Development / Sales
Selects Articles Sets inst-code and registration name
Order file
<order>.xml
Configuration
Config-autogenerator.xml
Configuration selects Projects and settings
Project Configuration
<project>.xml
Licence Generator uses Order file Configurations
Automatic Licence Generator
AutoLicGenerator.exe
Produce order file
produce licence file(s) or activation keys
Web Registration Engine
eShop Engine
Script
licence file for customer A and project X
myapp.lic
file with activation keys for customer A and project X
myapp_activation_key.txt
Distribution (e-mail, ...)
Customer A
Orders or registers Product
Get a new version

1.3 Command Line Parameters

The Automatic License Generator is completely command-line driven (XML input files) and is designed to operate without user interaction. You can develop an own shell (e.g. web based input pages) or use the Windows Scripting host.

Normally the workflow is as follows:

- Form for data input (e.g. shop system or registration system)

- Write the order file
- Call Automatic License Generator
- A licence file or Activation Key is generated
- Send the file or Key to the customer or provide a download

The Automatic License Generator is started as follows:

AutoLicGenerator.exe Parameter

The following parameters are possible:

Param.	**Description**		**Example**
-c	Alternate Product Configuration	If omitted**, config-autogenerator.xml** in the same directory as the AutoLicGenerator.exe is used	-c c:\test\myconfig.xml
-l	Write logfile	Set logging on or off *-l logfilename* -> activates logging *-l off* -> deactivates logging	-l logmyorder
-o	**The order file**	**This parameter is mandatory**	**-o myorder.xml**
-x	Output-Path	If omitted the generated licence files and Activation Key files are stored in the directory of the AutoLicGenerator.exe file	-x c:\output
-xl	Output-Licencefile	Path and name of the licence file that will be created	-xl c:\out\myapp.lic
-xk	Output-Activation Key file	Path and name of the Activation Key file that will be created	-xk c:\out\myapp.txt
-xml	Output as XML	The Activation Key files are written in XML format	-xml yes
-code x	Retrieves the installation code of the PC	x is the number of the installation code – e.g. code 2 = retrieve MAC address	-code x

Note

If you use a pathname then the pathname may not include umlauts as äüö.

Example: AutoLicGenerator.exe –o order.xml

1.3.1 Registry settings

LICENCE PROTECTOR Automatic License Generator provides you with an extensive error tracking. You can activate the error tracking via the registry or with the **command line parameter -l.** There are a lot of registry settings. You should modify only the listed ones.

HKEY_CURRENT_USER\Software\Mirage\AutoLicGenerator\**Log**

Name	**Description**
LoggingEnabled	0 = no error log is written 1 = Error log is enabled
LogLevel	256 = Extensive Log level. Recommended for error tracking
LogPath	Path for the log files.

The name of the log file is YYYY-MM-DD-HH.log

- YYYY = Year
- MM = Month
- DD = Day
- HH = Hour

A Logfile called 2005-02-14-09.log indicates that the application was **started** February 14th 2005 between 9 am and 10 am. As long as it runs the log output will be written in this file.

If you experience problems you can contact our support. They will probably ask for the log file(s) to analyze the problem. Send the files along with a detailed explanation to support@mirage-systems.de .

2 The Configuration Files

All configuration files are in XML format. If you want to use an umlaut (ö,ä,ü etc.) in an XML order or project file, then the XML file must be saved in UTF-8 format.

There are 3 configuration files:

- **The Order file** – it contains the data which are used as input to generate a licence file or a key – normally one or more products
- **The Product Configuration File** – it contains the definition on a ***product level*** what has to be generated for this product
- **The Project File** – this is the template of a licence file

If a setting could be defined in several configuration files the following rule is used:

- If a setting is made in the order file the corresponding settings in the product configuration file or the project file are overridden.
- If a setting is made in the product configuration file the corresponding setting in the project file is overridden.
- If no setting is made in order file or product configuration file the setting in the project file is used.
- If no setting at all is made in a configuration file then the default is used.

So there is the following priority:

Order file

overrides

Product Configuration File

overrides

Project file.

Note:

If you do not want a setting to be used in a higher configuration file then do not use that tag in the configuration file.

For example: You have the possibility to set the used method of CopyProtection in the *order file* or the **product configuration file**. If you set it in the *order file* the setting in the **product configuration file** is ignored. So if you want to use the setting in the **product configuration file** then omit the tag <CopyProtection> in the *order file*. Do not set <CopyProtection> to 0 in order to tell Automatic License Generator to ignore that value and use that value from the **product configuration file**.
The neutral value could only be set **by omitting the whole tag**. In the example by removing the line `<CopyProtection>0</CopyProtection>` from the order file.

2.1 The Order File

The order file is the input file with the customer and product data. The file maintains **all products** with quantities **for one customer**. It is not possible to include orders for several customers in one order file. It also holds information about the installation code if a copy protection is used.

2.1.1 The XML-Tags of the order file

All fields in black (marked with an *) are mandatory fields.

Tag	Description	Comment
`<LP-Order> *`	Surrounds the whole order definition	
`<OrderInformation> *`	Surrounds the header information of the order file	
`<OrderID>`	Info about the order. E.g. necessary in shop systems to store that specific order in a database	Unused by AutoLicGenerator
`<PurchaseDate>`	Info about the date of that specific order	Unused by AutoLicGenerator
`<NoOfKeys>`	Allows generating key lists. E.g. if you want to have 100 different keys (with the same attributes) set this value to 100.	Default is 1
`<KeyOutputStyle>`	A numerical value to control the output of a key in a key-file. Possible values are: 0: writes only the key 1: writes the key and describes problematic characters 2: writes the key and describes all characters. If this value is set then it overrides a value in the corresponding ***product configuration file***	
`<KeyTextLanguage>`	A numerical value to control the used language in a key-file. Possible values are all	If no KeyTextLanguage in the order file or

	language numbers (see list of languages) . If this value is set then it overrides a value in the corresponding ***product configuration file***	the product configuration file is set then the current language is used.
<Product> *	Header for product information	
<ProductID> *	The product reference to the product configuration. This value is mandatory	Link to a product in the **product configuration file**
<Quantity>	The quantity for that product	If omitted this value is 1
<AbsoluteLics>	A Yes/No Value. Only useful for key generation. If set to Yes the number of licences in a key is not added to the present number of licences in the licence file, it overwrites the number of licences with this absolute value. This value overrides a value <AbsoluteLics> in the corresponding ***product configuration file***	Default is not set. Example: In a licence file there are 5 licences for a Module X. A key for 3 licences with AbsoluteLics set to no will increase the number of licences to a total of 8 licences. The same key with AbsoluteLics set to yes will set the total number of licences to 3.
<Days>	Default time limitation – number of days. This value overrides a value < Days> in the corresponding ***product configuration file***	Corresponds to the field *valid number of days* in the License Generator
<AbsoluteDays>	A Yes/No Value. Only useful for key generation. If set to Yes the number of days in a key is not added to the present number of days in the licence file, it overwrites the number of days with this	Default is not set. If set to yes a key will set the maximum number of days to <Days> unattached if

	absolute value. This value overrides a value <AbsoluteDays> in the corresponding ***product configuration file***	there are already existing number of days.
<ValidUntilDay>	Default time limitation – valid until day. This value overrides a value < ValidUntilDay> in the corresponding ***product configuration file***	Corresponds to the field *valid until day* in the License Generator Use an explicit date in the format set in Windows control panel e.g. 5/16/2005 or use a term +X to calculate the date as today plus X days. E.g. +30 calculates a date 30 days from today.
<WebActivation>	Sets the WebActivation state of a module to one of the following values: 0 = none (default) 1 = required 2 = activated This value overrides a value in the corresponding ***product configuration file***	**For Serial Numbers it must be set to required**
<MakeSerial>	A Yes/No Value.	**For Serial Numbers it must be set to yes.** You do not need to specify this value as by default the definition <IsSerial> of the project template is used

`<ForceOnlineCheck>`	A Yes/No Value. Only useful for key generation. If set to yes the generated key can only be applied if online checked by the **Activation Server**. That additionally ensures that this key could only be applied one time. This value overrides a value in the corresponding ***product configuration file***	Default is not set. **For Serial Numbers it must be set to no.**
`<WebServiceURL>`	The URL where the **Licence Protector Activation Server** is located. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<ShowWAStartPage>`	Controls whether the start page of the form to connect to the **Activation Server** is displayed. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<ShowWAProgressPage>`	Controls whether the progress page of the form to connect to the **Activation Server** is displayed. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<ShowWAResultPage>`	Controls whether the result page of the form to connect to the **Activation Server** is displayed. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<SecurityLevel>`	Controls the security level of the licence file. Values are	Default is not set

	BASIC and ADVANCED. This value overrides a value in the corresponding ***product configuration file***	
`<TamperDetection>`	Controls whether the detection of manipulation is turned on or off. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<TamperDetectionMode>`	Controls the mode of the detection of manipulation. Allowed values are “auto”, “manual” and “off”. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<TamperDetectionGraceTimes>`	Controls the number of times the local Run Number may be greater than the Run Number in the licence file. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<LicenceVerification>`	Controls the verification mode for online checks. Allowed values are “on”, “off”, “frozen” and “deactivated”. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<EndVerification>`	Beginning from that date the ongoing online checks in CheckLicence are cancelled. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<AllowChangeWASURL>`	Controls whether the user may change the URL of the Web Activation service. This value overrides a value in the corresponding ***product configuration file***	Default is not set

`<AllowSetVal>`	Controls whether SetVal calls are allowed for that licence file. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<TagValue>`	The tag for the licence file. This value overrides a value in the corresponding ***product configuration file***	Default is not set
`<Registration> *`	Info about the customer and the manufacturer name	
`<RegistrationName> *`	The name of the customer in the licence file	Displayed in the field *Company* within the Licence Viewer
`<Manufacturer>`	A name of a manufacturer. Normally this is your company name. If omitted or <UseManufacturerFromOrder> in the ***product configuration file*** is set to no, the default manufacturer of the **project file** is used.	Displayed in the field *Manufacturer* within the Licence Viewer
`<CopyProtection>`	Which type of copy protection should be used (0 is none, 1 is VolumeID, 2 is MAC address, 3 = hostname etc.). If this value is set then it overrides a value in the corresponding ***product configuration file***	Default is not set.
`<InstallationCode>`	The installation code or licence file ID. **Mandatory if in the product configuration or in a project configuration a copy protection is activated.** You have to make certain that this type of installation code matches with the defined copy protection type	This is the installation code or the licence file ID displayed in the licence viewer

2.1.2 A sample order file

You find this sample on the installation directory – file myorder.xml.

08152003

18.12.2003

4712

3

30

4713

5

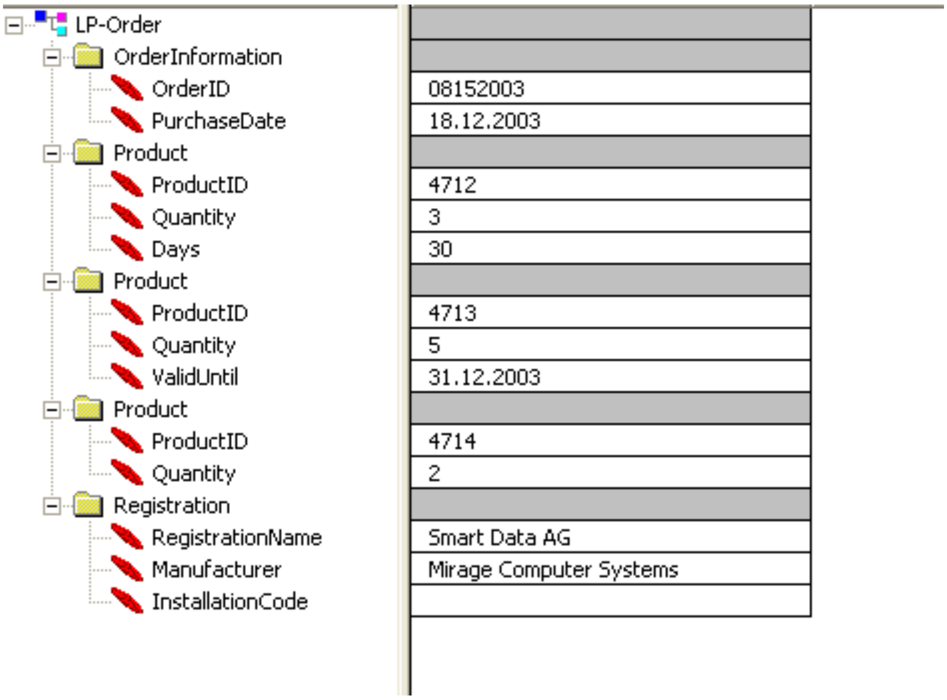
31.12.2003

4714

2

Smart Data AG

Mirage Computer Systems



LP-Order	
OrderInformation	
OrderID	08152003
PurchaseDate	18.12.2003
Product	
ProductID	4712
Quantity	3
Days	30
Product	
ProductID	4713
Quantity	5
ValidUntil	31.12.2003
Product	
ProductID	4714
Quantity	2
Registration	
RegistrationName	Smart Data AG
Manufacturer	Mirage Computer Systems
InstallationCode	

2.1.3 Field Mappings Order file ->Configuration file

The Automatic License Generator needs exactly the fieldnames described in the order file (e.g. ProductID, Quantity, InstallationCode). The fields are mapped to the corresponding fields in the licence file.

If your order processing system (e.g. element5) uses other fieldnames, you can map these fieldnames to the required names - **Note:** version 2.3 supports only mapping with the AutoLicGenElement5.exe file.

Example: Normally, the *installation code* is delivered from element 5 within the field *Additional1*. If you want to use the field *Additional2* for the installation code, you can use the field mapping.

These are the optional field-mappings

OrderID
PURCHASE_ID

RegistrationName1
REG_NAME

RegistrationName2
COMPANY

RegistrationName3
LASTNAME

RegistrationName4
EMAIL

Manufacturer
RESELLER

InstallationCode
ADDITIONAL1

ProductID
PRODUCT_ID

Quantity
QUANTITY

2.2 The Product Configuration file

The **product configuration file** defines **one or many** products / articles.

A product can consist of ***several modules***. To identify a product, a **product ID** is used. The product ID should be the article number in a shop system. Each module is refers with its **module ID** to the **project file**.

Example

The product *ERP Enterprise Edition* (product ID 4711) consists of the modules:

- D1001 (ERP Module)
- D1003 (Number of Employees)
- D1005 (Custom Reports)
- …….

The product ID **4711** is the article number in the shop system, where the module ID's D1001, D1003, D1005 are defined in the **project file.**

In the **project file**, the detail definition of the module (e.g. yes/no module, user module) is stored. **This must be the same project file that you use with the License Generator to generate the licence file manually.**

The **product configuration file** includes additional information about:

- Which is the project XML file for the article
- What should be generated? licence file or Activation Key(s)

The **product configuration file** can include any number of articles, which could use any number of **project files**. The number of **project files** is only limited by the licences you have bought.

The standard file name for the configuration file is **config-autogenerator.xml**, but the file name can be changed with a command line parameter (see chapter command line parameters).

Note

- If the configuration file is used for the Web Activation server then the **ProductID has to match the Module ID** of the module that has to be activated.

2.2.1 The XML-Tags of the product configuration file

All fields in black (marked with an *) are mandatory fields.

Tag	Description	Comment
<Connector> *	Surrounds the whole product configuration	
<Product> *	Surrounds a product definition including one or more modules. There have to be at least one Product-Tag in a valid **product configuration file**	
<ProductID> *	Defines the unique identifier of the product. With that value the product is **linked to the order file**. This information is mandatory	
<ProductName>	The name of the product e.g. "Christmas Edition"	Unused by AutoLicGenerator. Just a description of the product
<ProjectFilename> *	The **project file**name (with pathname). Mandatory.	Link to **project file**name with the module definition
<UseManufacturerFromOrder>	A Yes/No value to define whether the manufacturer name from the order file (if specified) is used or not	Default = Yes. Only necessary when creating a licence file
<CopyProtection>	Which type of copy protection should be used (0 is none, 1 is VolumeID, 2 is MAC address, 3 = hostname etc.). If this value is set then it overrides a value CopyProtection in the corresponding **project file**	You have to provide the installation code in the field <InstallationCode> in the order file when this field has a value other than 0.
<LicenceFile>	A Yes/No value to define that the output for that product should be a licence file. Note that products which are ordered together and which use the same ProjectFilename will be together in one licence file! (See below for further details).	Either <LicenceFile> or <ActivationKey> should be set to yes
<ActivationKey>	A Yes/No value to define that the output for that product should be a file containing Activation Keys. Note that all Activation Keys for products which are ordered together and which use the same ProjectFilename will be together in	Either <LicenceFile> or <ActivationKey> should be set to yes. The key is written into the file: namelicencefile_activation_key.txt unless you changed the name with the –xk switch.

	one file! (See below for further details)	
`<LicenceFileID>`	A Yes/No value to use the LicenceID for encoding the Activation Key. This key can only be applied, when the licence fileID has the same ID as the value provided in the field <InstallationCode> in the order file. CopyProtection is overridden to 0	You have to provide the licence file ID in the field <InstallationCode> in the order file
`<CreateCopy ProtectionKey>`	A Yes/No value to create a key for activating the CopyProtection. You have to set the CopyProtection value und you have to provide an InstallationCode	Default = No <ActivationKey> has to be set to yes to have a key output at all.
`<CreateCopy UnprotectionKey>`	A Yes/No value to create a key for deactivating the CopyProtection. You have to set the CopyProtection value und you have to provide an InstallationCode	Default = No <ActivationKey> has to be set to yes to have a key output at all.
`<KeyOutputStyle>`	A numerical value to control the output of a key in a key-file. Possible values are: 0: writes only the key 1: writes the key and describes problematic characters 2: writes the key and describes all characters. This value is overridden by a set value <KeyOutputStyle> in the order file.	Default is not set. Note that this style is ignored on key output in XML or in a list. Examples: 0: 1: 2:
`<WebServiceURL>`	The URL where the **Licence Protector Activation Server** is located. This value overrides a value < WebServiceURL> in the corresponding ***project file*** but is overridden itself, if the value <WebServiceURL> is set in the *order* file	Default is not set.
`<ShowWAStartPage>`	Controls whether the start page of the form to connect to the	Default is not set.

	Activation Server is displayed. This value overrides the value in the corresponding ***project file*** but is overridden itself, if the value is set in the *order* file.	
`<ShowWAProgressPage>`	Controls whether the progress page of the form to connect to the **Activation Server** is displayed. This value overrides the value in the corresponding ***project file*** but is overridden itself, if the value is set in the *order* file.	Default is not set.
`<ShowWAResultPage>`	Controls whether the result page of the form to connect to the **Activation Server** is displayed. This value overrides the value in the corresponding ***project file*** but is overridden itself, if the value is set in the *order* file.	Default is not set.
`<KeyTextLanguage>`	A numerical value to control the used language in a key-file. Possible values are all language numbers (see list of languages) . This value is overridden by a set value in the order file.	If no KeyTextLanguage in the order file or the product configuration file is set then the current language is used.
`<SecurityLevel>`	Controls the security level of the licence file. Values are BASIC and ADVANCED. This value is overridden by a set value in the order file.	Default is not set
`<TamperDetection>`	Controls whether the detection of manipulation is turned on or off. This value is overridden by a set value in the order file.	Default is not set
`<TamperDetectionMode>`	Controls the mode of the detection of manipulation. Allowed values are “auto”, “manual” and “off”. This value is overridden by a set value in the order file.	Default is not set
`<TamperDetectionGraceTimes>`	Controls the number of times the local Run Number may be greater than the Run Number in the licence	Default is not set

	file. This value is overridden by a set value in the order file.	
`<LicenceVerification>`	Controls the verification mode for online checks. Allowed values are “on”, “off”, “frozen” and “deactivated”. This value is overridden by a set value in the order file.	Default is not set
`<EndVerification>`	Beginning from that date the ongoing online checks in CheckLicence are cancelled. This value is overridden by a set value in the order file.	Default is not set
`<AllowChangeWASURL>`	Controls whether the user may change the URL of the Web Activation service. This value is overridden by a set value in the order file.	Default is not set
`<AllowSetVal>`	Controls whether SetVal calls are allowed for that licence file. This value is overridden by a set value in the order file.	Default is not set
`<TagValue>`	The tag for the licence file. This value is overridden by a set value in the order file.	Default is not set
`<RunNumberKeys>`	The RunNumberKeys in that section will be included in this product. Notice to have ActivationKey set on.	
`<AnonymizedMetrics>`	true	True / False. Default is true.
`<SignalState>`	on	off, on, offline, noautoresend. Default is on
`<SystemSignals>`	fist;acmm;acam;daam;damm;fivd;eist	List of metrics events. Default is none
`<BackupInterval>`	30	0 = off (default). Set the backup interval instead of using SetBackupInterval
`<WebRequestTimeoutInSec>`	30	empty or 0 = default. Set the timeout in seconds instead of using SetWebRequestTimeoutInSec
`<RunNumberKey>`	The section of one Run Number Key	

<Type>	Sets the type of the key to create. Allowed values are: - TamperDetectionOn - TamperDetectionOff - ResetLocalRunNumber - ResetTotalRunNumber	Default is not set Use TamperDetectionOn/Off to switch the detection of manipulation on or off. Use ResetLocalRunNumber to reset a local installation. Use ResetTotalRunNumber to reset a network installation.
<Active>	Yes/No. If a key should be generated you have to set this value to yes.	Default is yes
<ForceOnlineCheck>	Controls whether this key has to be checked using the Web Activation server	Default is no
<Days>	Only for ResetTotalRunNumber: Number of days beginning with applying the key, that an application will reset its local Run Number.	Default is 0
<Modules> *	The modules in that project which should be included in the licence file or for which an Activation Key should be generated	If you don't want to create a CopyProtection or CopyUnprotection key, a minimum of 1 module has to be defined
<Module> *	The frame for the module generation	n modules per product are possible
<ModuleNumber> *	The ModuleNumber or also called ModuleID. **It has to be the same ID then used in the project file**	A ModuleID is **limited to 6 characters** from A to Z and digits. Special characters and umlauts are not allowed. **ModuleIDs are not case sensitive**
<Demoversion>	A Yes/No value. Set's the field demoversion of the module to yes or no	Corresponds to the field *Demoversion* in the License Generator
<Value>	The number of licences or Yes/No for a yes/no-Module (module type 4) as a **default value**. If that value represents a number of licences it **is multiplied with the quantity** value of the corresponding product-entry in the	**If it is a Yes/No module then the value is mandatory and must be Yes or No.** Corresponds to the field *number of licences* in the License Generator

	order file. E.g. if the default value for the module is 5 licenses and the customer orders 4 products, then the value for the module is 20	**Note** **The <value> setting must be defined here. With this release (2.3) it is not possible to use the value from the project file.**
<CreateMultipleKeys>	A Yes/No Value. Uses the quantity value to create not one key but the number of keys specified in the quantity	If 5 keys should be generated for 5 single user installations with value=1 then use CreateMultipleKeys=yes. Otherwise 1 key with value 5 is generated (default). FixedValue has to be set to *No* (default).
<FixedValue>	A Yes/No Value. If the module value should **not be multiplied** with the quantity of the order set this field to Yes	Allows to generate a fixed number of licences per licence file
<AbsoluteLics>	A Yes/No Value. Only useful for key generation. If set to Yes the number of licences in Value is not added to the present number of licences in the licence file, it overwrites the number of licences with this absolute value. This value overrides a value < AbsoluteLics> in the corresponding ***project file*** but is overridden itself, if the value <AbsoluteLics> is set in the *order* file	Example: In a licence file there are 5 licences for a Module X. A key for 3 licences with AbsoluteLics set to no will increase the number of licences to a total of 8 licences. The same key with AbsoluteLics set to yes will set the total number of licences to 3. Default is not set.
<Days>	Default time limitation – number of days. This value overrides a value < Days> in the corresponding ***project file*** but is overridden itself, if the value <Days> is set in the *order* file	Corresponds to the field *valid number of days* in the License Generator
<AbsoluteDays>	A Yes/No Value. Only useful for key generation. If set to Yes the number of days in Days is not added to the present number of	If set to yes a key will set the maximum number of days to <Days> unattached if there are already existing number

	days in the licence file, it overwrites the number of days with this absolute value. This value overrides a value < AbsoluteDays> in the corresponding ***project file*** but is overridden itself, if the value <AbsoluteDays> is set in the *order* file	of days. Default is not set.
`<ValidUntilDay>`	Default time limitation – valid until day. This value overrides a value <ValidUntilDay> in the corresponding ***project file*** but is overridden itself, if the value < ValidUntilDay> is set in the *order* file	Corresponds to the field *valid until day* in the License Generator Use an explicit date in the format set in Windows control panel e.g. 5/16/2005 or use a term +X to calculate the date as today plus X days. E.g. +30 calculates a date 30 days from today.
`<WebActivation>`	Sets the WebActivation state of a module to one of the following values: 0 = none (default) 1 = required 2 = activated This value overrides a value in the corresponding ***project file*** but is overridden itself, if the value is set in the *order* file	**For Serial Numbers it must be set to required.**
`<MakeSerial>`	A Yes/No Value.	**For Serial Numbers it must be set to yes.** You do not need to specify this value as by default the definition <IsSerial> of the project template is used
`<ForceOnlineCheck>`	A Yes/No Value. Only useful for key generation. If set to yes the generated key can only be applied if online checked by the **Activation Server**. That additionally ensures that this key could only be applied one time.	Default is not set. **For Serial Numbers it must be set to no.**

	This value overrides a value in the corresponding ***project file*** but is overridden itself, if the value is set in the *order* file	

Note:

- The output will be **one licence file or one Activation Key file** for all products of the same **project file**
- All keys of one order with the same project file are stored in one key file. The name of the key file is: *namelicencefile*_activation_key.txt. Alternatively the name of the key file could be set with Command Line Parameter -xk
- In the key file above each key is the name of the module.
- If CreateCopyProtectionKey and Module Activation Keys are generated together in a product the *namelicencefile*_activation_key.txt holds **all of these keys**. You can only apply an Activation Key for a module **after** the Copy Protection is activated with the generated CopyProtection Key.
- If several products are mixed in one licence file then all products must have the same copy protection scheme
- We recommend, that you generate a licence file with all modules and not only with the modules ordered
- It is not possible to order the identical module of a project in different products within one order if the same licence file is used

Example:

- Product A of project *MyTextprocessor.xml* defines Module C as demo for 30 days.
- Product B of Project *MyTextprocessor* defines Module C as full version without expiration

A customer orders product A → no problem, he will get his Textprocessor
Another customer orders product B → no problem too
A third customer orders product A together with product B. He would get one licence file for his Textprocessor with which module C? The demo or the full version without time limitation?

Make sure that a customer can not select a combination of products that produce that case or make sure that a module only exists in one product. This case is not a problem if your customer can only choose one product!

2.2.2 A sample configuration file

You find this sample on the installation directory – file config-autogenerator.xml.

4711
ERP Basic Edition - Demo Version
demo.xml
no
Yes
no
0
no
no

D1001
Yes
30
2004-12-30

D1003
3

D1005
3

4712
… more modules and products

2.3 The Project file

The **project file** defines the project and its modules. This file is also used in the License Generator and described in detail in the online help.

Make sure, that you use exactly the same project file for the License Generator and for the Automatic License Generator.

3 Return Codes

The Automatic License Generator will end with the following exit codes:

Code	Description
0	Success, output files successfully written
1	No product in product configuration
2	Product configuration could not be read. Check for a XML error.
3	No Modules for an project found
4	Loading of module failed
5	Inconsistent product data
6	Inconsistent module data
7	Order file could not be read. Check for a XML error.
8	Inconsistent order data
9	Order process failed
10	Writing of files failed
11	Licence expired. Perhaps no of projects exhausted.

If you execute the Automatic License Generator in a shell you will see a detailed problem description.

4 Special Features

4.1 Output of a file with Activation Keys

There are several possibilities to generate one or more keys in a key file. In the following samples the Project file Demo.xml is used.

4.1.1 Generating a list of keys

If you need several keys of the same type you can generate such a list with Activation Keys using the parameter **<NoOfKeys>** Tag of the order file. This key list can be used if you sell your software via bookstores and you provide a key within the booklet or if an e-shop systems does not support the integration of a License Generator tool but you can upload a key list.

This Example will produce a list of 100 keys:

The order file:

```
<LP-Order>
   <OrderInformation>
      <OrderID>08152003</OrderID>
      <PurchaseDate>18.12.2003</PurchaseDate>
      <NoOfKeys>100</NoOfKeys>
   </OrderInformation>
   <Product>
      <ProductID>4713</ProductID>
      <Quantity>1</Quantity>
   </Product>
   <Registration>
      <RegistrationName>Smart Data AG</RegistrationName>
   </Registration>
</LP-Order>
```

The configuration file

```
  …
<Product>
      <ProductID>4713</ProductID>
      <ProductName>Analysis Module</ProductName>
      <ProjectFilename>demo.xml</ProjectFilename>
      <LicenceFile>No</LicenceFile>
      <ActivationKey>yes</ActivationKey>
      <LicenceFileID>No</LicenceFileID>
      <CopyProtection>0</CopyProtection>
      <Modules>
         <Module>
            <ModuleNumber>D1002</ModuleNumber>
            <Value>3</Value>
            <AbsoluteLics>True</AbsoluteLics>
            <ValidUntilDay>+30</ValidUntilDay>
            <WebActivation>1</WebActivation>
            <ForceOnlineCheck>Yes</ForceOnlineCheck>
         </Module>
      </Modules>
</Product>
```

This configuration will produce a list of 100 keys which all will set the module D1002 to a final expiry date (today's date +30) and enables the Web Activation flag of the module. The module licences are set to 3 (AbsoluteLics) no matter what number of licences are there before. Before it can be applied, every key has to be checked online in the Web Activation server (ForceOnlineCheck) see Documentation of Web Activation service)

The output is:

```
SAG2V-5CL93-9EgIV-3tR87-ZQU5k-68143;D1002
bMG2l-5CL93-9EgIN-3tR87-fQU5k-6D146;D1002
mhG2u-5CL93-9EgIB-3tR87-5QU5k-6w14k;D1002
kwG2d-5CL93-9EgIc-3tR87-1QU5k-6514e;D1002
7uG2W-5CL93-9EgIo-3tR87-0QU5k-6j14x;D1002
JvG2M-5CL93-9EgIY-3tR87-CQU5k-6j14t;D1002
4hG2F-5CL93-9EgIw-3tR87-GQU5k-6J14M;D1002
LHG2b-5CL93-9EgIQ-3tR87-HQU5k-6814Z;D1002
…
```

Or if the `-xml yes` option is used then

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<ALG>
   <TheKey>ING2V-5CL93-9EgIM-3tR87-5QU5k-6J14X</TheKey>
   <TheKey>x8G2J-5CL93-9EgIo-3tR87-NQU5k-6q14W</TheKey>
   <TheKey>OYG2x-5CL93-9EgIA-3tR87-oQU5k-6Y143</TheKey>
…
   <TheKey>PiG22-5CL93-9EgI7-3tR87-LQU5k-6b14t</TheKey>
   <TheKey>X8G2w-5CL93-9EgIC-3tR87-GQU5k-6F14T</TheKey>
   <TheKey>NOG2C-5CL93-9EgIu-3tR87-XQU5k-6s14w</TheKey>
</ALG>
```

Note:

- When using NoOfKeys greater than 1 then Key Output style is ignored. You simply get a list of keys
- Only 1 key type per list is possible – e.g. an copy protection key and a standard key in one key list is not possible

4.1.2 Using XML output

If calling AutoLicGenerator with the –xml yes parameter the key output is done in XML format:

Example: `AutoLicGenerator myorder.xml` **`-xml yes`** will result in

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<ALG>
   <ActKey>
      <KeyType>0</KeyType>
      <TheKey>a3G2p-5CL93-9EgII-3tR87-GQU5k-6F14h</TheKey>
      <ModuleID>D1001</ModuleID>
      <Modulename>ERP Module</Modulename>
   </ActKey>
</ALG>
```

4.1.3 Using extended key descriptions

There are three key output styles available:

- Style 0: Only the key
- Style 1: The key and a description of all problematic characters
- Style 2: The key and a description of all characters

This extended key description can be used if the key is intended to be sent via fax and it is expected to be blurred. The customer has difficulties to read the key (is it a zero or an O?)

Style 0 will generate the output:

```
Key for module ERP Module:
RWG2v-5CL93-9EgIt-3tR87-dQU5k-6M140
```

Style 1 will generate the output:

```
Key for module ERP Module:
lYG25-5CL93-9EgIA-3tR87-uQU5k-6U14i
```

```
In block 1 at position 1 there is the small letter l
In block 3 at position 4 there is the capital letter I
```

Style 2 will generate the output:

```
Key for module ERP Module:
xPG2H-5CL93-9EgIe-3tR87-CQU5k-6r14g

In block 1 at position 1 there is the small letter x
In block 1 at position 2 there is the capital letter P
In block 1 at position 3 there is the capital letter G
In block 1 at position 4 there is the digit 2
In block 1 at position 5 there is the capital letter H
In block 2 at position 1 there is the digit 5
In block 2 at position 2 there is the capital letter C
In block 2 at position 3 there is the capital letter L
In block 2 at position 4 there is the digit 9
In block 2 at position 5 there is the digit 3
In block 3 at position 1 there is the digit 9
In block 3 at position 2 there is the capital letter E
In block 3 at position 3 there is the small letter g
In block 3 at position 4 there is the capital letter I
In block 3 at position 5 there is the small letter e
In block 4 at position 1 there is the digit 3
In block 4 at position 2 there is the small letter t
In block 4 at position 3 there is the capital letter R
In block 4 at position 4 there is the digit 8
In block 4 at position 5 there is the digit 7
In block 5 at position 1 there is the capital letter C
In block 5 at position 2 there is the capital letter Q
In block 5 at position 3 there is the capital letter U
In block 5 at position 4 there is the digit 5
In block 5 at position 5 there is the small letter k
In block 6 at position 1 there is the digit 6
In block 6 at position 2 there is the small letter r
In block 6 at position 3 there is the digit 1
In block 6 at position 4 there is the digit 4
In block 6 at position 5 there is the small letter g
```

4.1.4 Set Language

If you need to generate Key files for different languages you can use XML-Tag KeyTextLanguage. This example will produce a Key file with Style 1 output in French, regardless what language is set on your machine:

```
<LP-Order>
   <OrderInformation>
      <OrderID>08152003</OrderID>
      <PurchaseDate>18.02.2005</PurchaseDate>
      <KeyOutputStyle>1</KeyOutputStyle>
      <KeyTextLanguage>30000</KeyTextLanguage>
   </OrderInformation>
   <Product>
      <ProductID>MyDemo</ProductID>
      <Quantity>1</Quantity>
   </Product>
   <Registration>
      <RegistrationName>Smart Data AG</RegistrationName>
      <Manufacturer>Mirage Computer Systems</Manufacturer>
      <InstallationCode>12345678</InstallationCode>
   </Registration>
</LP-Order>
```

This will produce the following output:

```
Clé pour module ERP Module:
ERG2I-5CL93-9EgIq-3tR87-OQU5k-6P14K

Le bloc 1 indique à la position 5 la lettre majuscule I
Le bloc 3 indique à la position 4 la lettre majuscule I
Le bloc 5 indique à la position 1 la lettre majuscule O
```