

ClassifyIt 7.1

ADMIN GUIDE

ClassifyIt, an Add-In for Outlook, Word, Excel, PowerPoint

support@ugarbe.de
2026

Table of Contents

Article I. Overview 5
(a) Preparation Phase 6
(b) Deployment Phase 7
Article II. Installation 9
Article III. Initialisation Process 11
Article IV. Prepare ClassifyIt 13
Section 4.01 Configuration of ClassifyIt.json 13
Section 4.02 Integrity of Configuration-File (ClassifyIt_ConfigSigner) 14
Article V. Configuration 15
Section 5.01 Introduction 15
Section 5.02 Registry and Configuration File 15
Section 5.03 Registry Keys 16
(a) Description: 16
(b) Example: 16
Section 5.04 Configuration File 17
Article VI. ClassifyItUserData.json 36
Section 6.01 "categories" 36
Section 6.02 "keywords" 36
Section 6.03 "templates" 36
Section 6.04 "ECC" 36
Section 6.05 "quickSelect" 37
Article VII. Manual Installation 38
Section 7.01 Files 38
Section 7.02 Procedure 38
Section 7.03 Explorer Integration 40
Article VIII. Manual Deinstallation 41
Article IX. Classification Properties (X-Header Information) 43
Article X. Data Leak Protection (DLP) 44
Article XI. Prevent Disabling of ClassifyIt 45
Section 11.01 Prevent ClassifyIt being disabled by a user 45
Section 11.02 Ensure ClassifyIt is loaded at all times 46
Article XII. Event Logging 47

Section 12.01 Windows Event Log 47
Section 12.02 Limits 48
Article XIII. Information Manager 49
Section 13.01 Classifications 50
Section 13.02 Release Markings 51
Section 13.03 Release Groups 51
Section 13.04 Quick Select 52
Section 13.05 Default Marking 53
Section 13.06 Formatting 53
Article XIV. CI-TDF Containers 54
Section 14.01 Signatures and Trust 54
Section 14.02 Saving Back 54
Section 14.03 STANAG 4774 Confidentiality Label 54
Article XV. iCLASS 56

ClassifyIt 7.1 – Administrator Guide

This document is for administrators to manage and distribute ClassifyIt. It contains installation aspects to understand how ClassifyIt behaves during boot time and where it expects its configuration parameters to enforce your classification markings to Office documents.

The reader should start with the Overview which provides a step-by-step best practice guide of configuring and deploying ClassifyIt linking to the important chapters. Many details in later chapters are of less importance for a typical deployment, as support is given by support@ugarbe.de, for initial installations and later if subscription is active.

ClassifyIt is an Add-In for Outlook, Word, PowerPoint and Excel of the Microsoft Office 2010 Suite and later products, and can integrate in the File Explorer (Shell). The main testing has been conducted on the Office 2019 Suite. Compatibility should be with higher versions from Office 2010 onwards. Office 365 and Microsoft 365 versions should be compatible provided the Windows based version is installed (web-based versions will not work).

ClassifyIt is SHAREWARE software and must be licensed when used after the test-period. The normal test-period is 3 months however, longer test-periods can be granted if requested. Please contact us at: support@ugarbe.de

Before deploying ClassifyIt, conduct thorough technical integration and user testing in your reference environment. Focus should be given to add-in integration and potential incompatibilities with other add-ons.

Article 1. Overview

The following describes a typical best practice installation path of ClassifyIt. Whenever problems or questions occur, please contact your support team at: support@ugarbe.de

Please note that most details of this document are not required by your team for a typical ClassifyIt deployment. The deployment is an interactive process between your administrator and support@ugarbe.de. The deployment is straightforward and requires an administrator of your Windows and/or Microsoft Office installations and for the security markings an information management expert (or someone who understands the classification markings of your company/organisation).

In any case – test ClassifyIt thoroughly before deploying it operationally.

(a) Preparation Phase

The preparation has the aim of establishing the base configuration of your ClassifyIt implementation and consider how to deploy ClassifyIt to your users.

During this phase **technical administrator** knowledge and **information management** knowledge is necessary.

#	Who	What
1	Administrator	Familiarise with installation details: - Article II *Installation* - Article III *Initialisation Process*
2	Information Manager	Provide details of classifications: See Article XIII Information Manager - Provide some examples of classification markings; - Provide list of classifications; - Provide list of release markings (optional); - Consider release groups (optional); - Provide default classification and release marking (this will be added to a new document/email by default); - Consider the most commonly used classification and release markings (used for quick select);
3	Administrator	Consider Data Leakage Protection (DLP) options: - ClassifyIt encodes the classification and release marking of each email in the X-Header and for documents in a custom property. Gateway and mail-servers can set inspection rules to allow, reject or log the sending of emails or copy-pasting of documents. As such you could reject sending higher classified information without encryption, or restrict the distribution without release markings, etc.
4	Administrator	Finalise the configuration: - Contact support@ugarbe.de with the details from the Information Manager (#2); - ugarbe.de will provide you with the configuration file; - Test the received configuration and coordinate with the Information Manager if things need to be changed/updated; - Finalise the configuration in coordination with support@ugarbe.de;

(b) Deployment Phase

The administrator needs to decide how to deploy ClassifyIt to the users' workstations. It is recommended to use the same deployment as used for the Microsoft Office suite.

Running the ClassifyIt setup program installs ClassifyIt on the workstation where it is executed. Run interactively, it asks for the operational readiness statements to be confirmed. A silent installation confirms them with /ACCEPTREADINESS on the command line - without it, the setup refuses to install silently.

ClassifyIt loads OpenSSL (libcrypto-3-x64.dll, for 32-bit Office libcrypto-3.dll) only from its own program folder, by its full path - never a copy another program has placed in the PATH or in its own folder. If the file is missing, ClassifyIt reports it and does not encrypt, decrypt or sign; reinstall ClassifyIt. libssl is not used.

Text files from a CI-TDF container (.txt, .csv, .json, .xml, .md and similar) open in the editor Windows has for them. Many editors (Windows 11 Notepad, Notepad++) keep unencrypted copies of open files for their next start; ClassifyIt warns about this when such a file is opened, with that editor's settings to turn it off. A user who chooses "don't show this again" sets the DWORD HideTextEditorWarning = 1 under HKEY_CURRENT_USER\Software\ClassifyIt.

The setup checks for Office 2016 or newer (Microsoft 365, 2019, 2021, 2024 - all version 16.0) in its own bitness: the 64-bit setup needs 64-bit Office, the setup for 32-bit Office needs 32-bit Office. Office of the other bitness stops the setup, also a silent one, because Office would never load the add-in. Without Office 2016 or newer the setup asks whether to install anyway; a silent setup goes on. /SKIPOFFICECHECK skips the check, e.g. when Office is installed afterwards. The result is in the setup log.

In a bigger environment, ClassifyIt is distributed with the software distribution already used for Microsoft Office, for example Microsoft Configuration Manager (formally MECM or SCCM). The distribution either runs the setup program silently, or carries out the steps of Manual Installation and Manual Deinstallation as scripts:

Application with the setup program: the setup program as the content, with the install and the uninstall command (the uninstaller is created in the program folder):

```
"setup-ClassifyIt v7.1(64-bit)(<customer>).exe" /VERYSILENT
/SUPPRESSMSGBOXES /NORESTART /ACCEPTREADINESS /LOG="%TEMP%\ClassifyIt-
setup.log"
```

```
"C:\Program Files\ClassifyIT\unins000.exe" /VERYSILENT /SUPPRESSMSGBOXES
/NORESTART
```

Application with scripts: the program files of one build as the content, an install script with the steps of Manual Installation, an uninstall script with the steps of Manual Deinstallation.

Both run in the system context with administrator rights (MECM deployment type: Installation behavior "Install for system", whether or not a user is signed in).

Detection: the file version of ClassifyIt64.dll in the program folder, for example "C:\Program Files\ClassifyIT\ClassifyIt64.dll" with a version greater than or equal

to 7.1.0.0 (MECM detection method: File System).

Office closed: the add-in (ClassifyIt64.dll, ClassifyIt32.dll) can only be replaced while no Office application uses it. In MECM, list WINWORD.EXE, EXCEL.EXE, POWERPNT.EXE, OUTLOOK.EXE and ClassifyItTool.exe on the Install Behavior tab of the deployment type, so that the users are asked to close them first - or deploy in a maintenance window. A silent setup does not wait: the ClassifyIt background tool is stopped after 20 seconds, and files that Office still uses are replaced at the next restart of Windows.

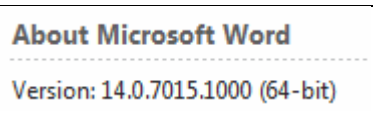
Per user: the Explorer integration is set up for every user by Windows Active Setup at their next sign-in (step 5 of Manual Installation) - the deployment does not have to run as the user. Run as SYSTEM, the setup leaves it to Active Setup.

Configuration: the setup links the configuration file in its program folder. The registry values for other locations (step 3 of Manual Installation) and those of Prevent Disabling of ClassifyIt are set by the script, or by Group Policy (Group Policy Preferences, Registry). The configuration file itself is best kept in two locations on network shares or web servers, so that it can be changed without a new deployment.

Updates: a new version is deployed as a new revision of the application, or as a new application that supersedes the old one. The setup stops the ClassifyIt background tool, and each new version of it raises the Active Setup "Version", so that the Explorer integration is renewed for every user; an install script does the same (step 1 of Manual Deinstallation, and step 4 of Manual Installation).

Article II. Installation

Before installing ClassifyIt, please ensure that the version of Microsoft Office and ClassifyIt match. Microsoft Office can be installed as a 64-bit version or 32-bit version. **ClassifyIt has to be of the same version as your Office installation**, otherwise ClassifyIt will not be able to be executed! To verify, open any of the Office applications, like Word, and do the following:

In Office 2010 verify with: *File → Help*	
In Office 2016 verify with: *File → Account → About Word*	

Before installing ClassifyIt, please ensure that all Office applications are closed to permit ClassifyIt to properly register as an Add-In to Outlook, Word, Excel and PowerPoint. The registration could fail if any of the applications is still running. If you encounter problems please check with the Windows TaskManager if really all Office applications are closed, and if required end the application through the TaskManager – afterwards reinstall ClassifyIt.

Start the installation by executing the provided setup program.

During the installation process of the 64-bit version, the ClassifyIt Add-In application and configuration files are installed in the %PROGRAM FOLDER%\ClassifyIt directory (e.g. C:\Program Files\ClassifyIt). This is the program directory of the setup program. For mass deployment using other deployment means, the directory can be different.

After the installation the Add-In is directly usable, however the configuration file (default: ClassifyIt.json) must be changed to support the classification requirements of your organisation/company. If help is needed, please see section Configuration or contact support@ugarbe.de.

After the installation the ClassifyIt toolbar is available on all open or newly created emails, Word documents, Excel spreadsheets and PowerPoint presentations.

The 7.1 setup installs the Windows Explorer integration together with the add-in: ClassifyItTool.exe (exporting, opening and saving back CI-TDF containers, Certificate Management) and ClassifyItShell.dll (the classification stamp on file icons). ClassifyItTool.exe runs ClassifyIt64.dll from its own folder, so all files must always come from the same build.

The Explorer integration is set up per user: for the user who installs right after the installation, for every other user at their next sign-in - Windows Active Setup runs ClassifyItTool.exe /register once per user, and again after an update.

Updating while ClassifyIt is in use: the setup stops the ClassifyIt background tool, asks the user to save their work and close Word, Excel, PowerPoint and Outlook (Retry, Ignore or Abort), and restarts Windows Explorer if it holds a ClassifyIt file - the open folder windows are opened again afterwards. A file that is still in use, for example in another user's session, is replaced for the next start of that program; its old version is deleted at the next restart.

Uninstalling does the same before it removes the files. The Explorer integration is removed for every user of the computer: for users who are not signed in, the uninstaller loads their registry from their profile for the moment. In the session of another signed-in user, Explorer keeps the icon handler loaded until that user signs out; the file is then deleted at the next restart.

The setup and the program files are not code-signed yet, so Windows shows the setup's publisher as unknown.

Article III. Initialisation Process

The ClassifyIt Add-In is a library which runs in the Microsoft Office context, as such it is programmed as a Dynamic Link Library (DLL). After the installation (running the setup executable) this add-in is in the following file:

%PROGRAM FOLDER%\ClassifyIt\ClassifyIt64.dll (32-bit Office: ClassifyIt32.dll)

Note: it is possible to change the folder and the name of the DLL – in this case the registration of the add-in DLL needs to be done using the 'regsvr32' Windows command.

Note: with v7 OpenSSL libraries are part of the installation package. Those two following dll files must be in the same folder as the Classifyit.dll:

libcrypto-3-x64.dll, libssl-3-x64.dll

When the ClassifyIt Add-In is started (within the Office context) it needs to load its configuration file, which is formatted in a JSON syntax – by default the configuration file is the following file:

%PROGRAM FOLDER%\ClassifyIt\ClassifyIt.json

The file-name and location of the configuration file is set by the Registry key HKEY_LOCAL_MACHINE\SOFTWARE\CLASSIFY_IT\ConfigFile – the key can be a URL or file-system path. For redundancy an alternate path to the configuration file should be provided through AlernateConfigFile.

When the ClassifyIt Add-In starts for the first time (e.g. by starting Outlook, Word, Excel or PowerPoint) the following files are created under the users CSIDL_APPDATA folder (e.g. C:\Documents and Settings\username\Application Data\Roaming):

- ClassifyItUserData.json

  Used for ClassifyIt User Data

- ClassifyIt.log

  Used for logging information of ClassifyIt

- ClassifyIt.json

  local copy of the configuration file. This file is overwritten whenever the applications are started (Outlook, Word, Excel, PowerPoint) and the central configuration file is available.

- Certificates

  folder storing user's digital certificate (private and public key) and recipients public keys, needed for the CryptIt module.

Containers opened from Windows Explorer or imported in Office are decrypted into %LOCALAPPDATA%\cli_temp_import, a local folder that does not roam with the user profile. Next to each decrypted file a hidden note, <file name>.citdf, records the container it came from, so that saved changes go back into it. The note also records the file times of the copy and of the container at the last save-back, so ClassifyIt knows whether the copy holds changes that are not in the container. Both are deleted when no program uses the file any more, and at sign-out - except a copy with such changes: it stays until the user saves it into a container, keeps it or discards the changes, and is offered again when Word, Excel or PowerPoint starts (their file types) or ClassifyItTool starts (all other types).

Article IV. Prepare ClassifyIt

Note: For a typical best-practice installation the details of this chapter are not important, as ugarbe.de provides all input. This chapter is a technical reference for those who procured the signature tool and want/have to configure ClassifyIt themselves.

After the installation there are 2 steps which need to be done. First the configuration file ClassifyIt.json needs to be adjusted to the required classification schema and second, the updated configuration file must be signed. Both steps are to be done by an administrator. The adjustment of the configuration file is most critical, contact support@ugarbe.de whenever in doubt or if problems are encountered.

Section 4.01 Configuration of ClassifyIt.json

The adjustment of the ClassifyIt.json configuration file is the most critical step in your deployment of ClassifyIt. The syntax of the JSON file is sensitive and must be absolutely correct, otherwise ClassifyIt will not work properly. The configuration, based on your requirements, will typically be done by ugarbe.de. The JSON file is signed to ensure its integrity, so that no-one else can change your classification schema. **If you do the configuration yourself, please ensure you have the signature tool procured.**

To make your own configuration file you could use the following guidelines:

a) understand your classification schema;

b) use an editor which understands the JSON format, for instance Notepad++ with the JSON Viewer plugin;

c) start with the original configuration file and adjust it in small steps to understand the changes;

d) most Office applications must be restarted to read the new configuration file. But in Outlook there is the possibility to use the admin_mode, then in Outlook the configuration file can be reloaded through an extra button in the menu;

e) check the JSON format with a tool (see b);

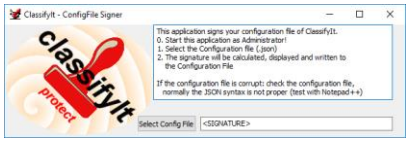
f) sign the JSON file (see next section)

g) test the configuration thoroughly.

Section 4.02 Integrity of Configuration-File (ClassifyIt_ConfigSigner)

The ClassifyIt.json configuration-file has a tamper protection to ensure its integrity through a signature, so that only authorised persons can change the configuration. The signature value is verified by the ClassifyIt plug-in when the configuration-file is loaded. The signature value must be calculated by the **ClassifyIt_ConfigSigner.exe**. This signing application is usually only used by ugarbe.de. Only in exceptional cases it is useful for the customer to procure the signer tool.

1. Start the ClassifyIt_ConfigSigner application (the application might have to be executed with administrator privileges, if the configuration file is in a protected folder, like a program folder).



2. At startup, load the authorised signing private key and its matching certificate. The signing key must match the configuration-verification certificate embedded in the deployed ClassifyIt add-in; an arbitrary certificate will not be accepted. Agree key provisioning with support@ugarbe.de. Keep the private key and its password restricted to authorised administrators, maintain a protected recovery backup, and coordinate any replacement with an update of the verifier in the add-in.
3. Select the configuration-file.
4. The configuration file gets properly signed and a success message is shown.
5. If signing fails, check the reported error, JSON syntax, signing key/certificate pair, password and file permissions. Correct the cause and sign again. Test the signed configuration with the deployed add-in before rollout.

Article V. Configuration

Section 5.01 Introduction

The proper configuration of the tool is the most critical and requires most attention during the deployment of ClassifyIt.

Section 5.02 Registry and Configuration File

ClassifyIt is controlled by a Registry Key and a Configuration-File:

1. Locator of Configuration-File
   1. Registry Key: **HKLM\SOFTWARE\CLASSIFY_IT** [holds two values with the path or URL and name to the Configuration-File]
      Note: for the 32-bit version of ClassifyIt the key is stored at: ***HKLM\SOFTWARE\Wow6432Node\CLASSIFY_IT***
      1. **ConfigFile**: [first try of ClassifyIT Add-In to load the Configuration-File]
      2. **AlternateConfigFile**: [alternative if file specified by ConfigFile does not exist, is not reachable or is corrupt]

1) Configuration-File
   <**classifyIt.json**> [holds all configuration parameters of ClassifyIt in a JSON formatted file]

Section 5.03 Registry Keys

HKLM\SOFTWARE\CLASSIFY_IT\

Two keys specify the fully qualified file-name of the <classifyIt.json> file.

ConfigFile=<PATH\FILE_NAME> or <URL>

AlternateConfigFile=<PATH\FILE_NAME> or <URL>

(a) Description:

The ConfigFile key specifies the preferred location of the <classifyIt.json> file. If the preferred file is not available (or corrupt) the alternative configuration file will be loaded during the start of Outlook, Word, PowerPoint or Excel. The configuration file could be on a local or shared drive, or on a web-server where the URL is to start with 'HTTP://'.

If no configuration file is available, ClassifyIt will try to load a previous copy of the configuration file in the user's CSIDL_APPDATA folder.

If none of the files is available then an error message is displayed to the user. If any of the files is not available or corrupt, an error/warning is written in the classifyit.log file.

Note: A configuration file is corrupt if either the JSON syntax is incorrect, or if the integrity of the file (provided through ClassifyIt_ConfigSigner.exe) is incorrect.

(b) Example:

```
ConfigFile='http://classifyit.eu/ClassifyIt.json'
AlternateConfigFile='z:\configData\classifyit.json'
```

Section 5.04 Configuration File

The <classifyIt.json> file controls content and behaviour of the ClassifyIt Add-In.

Note: The standard setup uses the filename: “ClassifyIt.json”, however any other name can be used as specified by the Registry (see Section 5.03).

The following describes the Format and then the different sections within the configuration file.

(i) Encoding

Save configuration files as UTF-8. Existing UTF-16 little-endian files with a byte-order mark can also be read.

Use a Unicode-aware JSON editor so that special characters remain correct, and re-sign the configuration after every change.

(ii) Format

The configuration file is JSON formatted (www.json.org/). This is a very efficient, powerful, text-based and human readable format.

It is important that the syntax of the configuration file is fully JSON compliant. Use a syntax checking editor to modify or generate the configuration file (e.g. Notepad++, notepad-plus-plus.org/, with a JSON plug-in, nppjsonviewer).

(iii) "classifyit_control"

Is a record of values controlling the overall behaviour of ClassifyIt:

```
"admin_mode"      <boolean>   specifies if in Outlook the
            configuration file can be reloaded, which might be
            useful during the creation of the configuration file
            by the administrator.
"use_default_marking"   <Boolean>   specifies if in Outlook a
            default marking should be put on emails when
            generated. The other applications automatically add
            the default marking when a new document is generated.
"default_marking"       <RECORD>    specifies the default
            classification and release marking assigned to new
            documents/emails, using the same "class" and "rel"
            fields described below (the same convention used
            throughout this file - see "classifications",
            "release_markings" and "quickSelect").:
            "class"     <STRING>    classification value which
                  must match one of the "class" values in the
                  "classifications" section.
            "rel" <ARRAY of STRING> release marking value(s) which
                  must match "rel" values in the
                  "release_markings" section; use an empty array
                  [] if no release marking applies.
```

"originator" <STRING> (new in ClassifyIt 7.0) short originator code (e.g. an ISO 3166 alpha-3 country code, or an organisational code/name). This will be embedded as the classification assertion's "org" value when exporting/encrypting a document to a CI-TDF container. If not specified, a fixed default is used. (Part of iCLASS standard).

"propertyKeyName" <STRING> (change from Classifyit 7.0 onwards) the name of the key in document properties and email x-header, which stores the classification-release value. ClassifyIt is optimised for the "iCLASS" standard, which is the default value. ClassifyIt will still use the iCLASS standard to represent classification and release.

"logEvents" <boolean> (new in ClassifyIt 7.1) records the reads, saves, exports and Outlook sends of classified items in the Windows event log (see Event Logging). Default: false.

Example:

```
"classifyit_control":
       {"admin_mode":         true,
       "use_default_marking": true,
       "default_marking": {"class": "R", "rel": []},
       "originator":          "ugarbe.de",
       "propertyKeyName":     "iCLASS",
       "logEvents":     true}
```

(iv) "version_control"

This object defines the date, time, version number and signature of the configuration file. The signature is generated by ClassifyIt_ConfigSigner and verified when the configuration is loaded (see Section 4.02).Section 4.02

"date" <STRING> date of the configuration file.

"time" <STRING> time of the configuration file.

"versionNumber" <STRING> version number of the configuration file.

"signature" **<STRING> contains the Base64-encoded configuration signature generated by ClassifyIt_ConfigSigner.exe. Recalculate it whenever the configuration changes. The example below contains a placeholder, not a valid signature; sign the completed configuration before use. The signing tool and authorised private key are available separately and must be restricted to authorised administrators.**

Example:

```
{"date":          "2026 10 21",
 "time":          "220415",
 "versionNumber": "7.0",
  "signature":"REPLACE_WITH_GENERATED_BASE64_SIGNATURE"}
```

(v) "classifications"

Is a list of objects which define each available classification. Each classification is defined through the following attributes:

"real"	**<STRING> the real classification name as it will appear on the document or email when selected, and used as the fallback label if "display" is not specified.**
"display"	<STRING> (optional) the caption shown for this classification in the ReleaseForm's classification picker; if not specified, "real" is used instead.
"additional"	<STRING> text which might be required in addition to a classification.
"trans"	<STRING> the translated classification name as it will appear on the document (email) when the classification is selected and when a translation is required (controlled through "requiresTranslatedLabel" of the "classification_formatting" settings). Translation should be used for an interoperable iCLASS label (used in default configurations).
"color"	<STRING> the color value (RGB) of the classification.
"class"	<STRING> the classification code stored in the marking property (replaces the legacy "x-class" name; the property name itself is set by "propertyKeyName" in the "classifyit_control" section). By default the iCLASS representation should be marked here to achieve interoperability.
"descr"	<STRING> a description of the classification which will be shown in the ClassifyIt classification menu as a 'hint'.
"enabled"	<BOOLEAN> defines if the classification can be selected by the user – this is normally true.
"icon"	<STRING> (optional) stamp for this classification on Explorer icons and previews and in the Quick Select menu – possible values are: amber, black, blue, darkgreen, darkred, green, grey, lightgrey, magenta, navy, orange, purple, red, teal, white
"class-level"	<INTEGER> (new with version 5.8) defines the classification level of the classification. In earlier versions the classification level was given by the order of the classifications – first lowest level, … last highest level. Now class-level defines this – lower number is a lower classification level and higher number is a higher classification level.
"sensitivity"	<STRING> corresponding Outlook sensitivity (Normal=0, Personal=1, Private=2, Confidential=3). The outlook_control setSensitivityLevel value defines if the sensitivity level is actually applied to the email.

Note: the classifications with the iCLASS codes P and U carry no release. The Release Form greys out the release options for them, and a release configured for them in "default_marking" or "quickSelect" is ignored; a warning is then written to ClassifyIt.log.

Explorer: a file whose class code is in the configuration shows that classification's "icon" (without "icon", a dot in its "color"), whoever classified it. A class code the configuration

does not have is never shown as another class: it gets a rounded square with an "i" in the colour of that class on classifyit.eu/iclass - P #171717, U #59636E, O #237A43, R #2463A6, C #D99A00, S #BB2635, T #650D20.

Files on network shares and mapped drives get their stamp too. Explorer first shows the plain icon; the file is read in the background and the stamp follows, so a slow or unreachable server never blocks the folder window. Files that are only in the cloud (OneDrive placeholders, offline files) are not downloaded for this. To turn network reading off - e.g. for slow WAN links - set the DWORD "NetworkFiles" = 0 under HKEY_LOCAL_MACHINE\Software\ClassifyIt\ShellIcons (for all users; it wins) or under the same key in HKEY_CURRENT_USER. Users switch their own setting with ClassifyIt > Hide / Show stamps on network drives in Explorer (ClassifyItTool networkstamps); a machine-wide value cannot be changed there.

Example:

```
 [{"real":  "WHITE",
 "display": "WHITE",
 "trans":   "Unclassified",
 "color":   "0x000001",
 "class":   "U",
       "descr":   "information can be distributed without restrictions
(subject to copyright restrictions)",
       "enabled": true,
       "class-level":   1,
        "sensitivity":"0",
  "icon":"black"}]
```

(vi) "release_markings"

Is a list of objects which define each available release marking. Each release marking is defined through the following attributes:

"real"	<STRING> the real release marking as it will appear on the document (email) when selected.
"trans"	<STRING> the translated release marking as it will appear on the document (email) when a translation is required (controlled through "requiresTranslatedLabel" of the "classification_formatting" settings).
"rel"	<STRING> the release code stored in the marking property (replaces the legacy "x-rel" name, which is still accepted as a fallback if "rel" is absent). For national release use of ISO 3-letter codes is advised (iCLASS Standard)
"descr":	<STRING> a description of the release marking which will be shown in the ClassifyIt classification menu as a 'hint'.

Example:

```
{"real":    "MANAGEMENT",
 "trans":   "MAN",
 "rel":     "MAN",
 "descr":   "Senior Management Staff of Company"}
```

(vii) "release_groups"

Is a list of objects which define groups of release markings. Each release marking group is a collection of release_markings. Each release marking group is defined through the following attributes:

"display"	<STRING> the display name for the group.
"group"	[<STRING>] (list of strings) each value must be one of the "rel" codes of the "release_markings" section (not the "real" display name). The group then represents all the corresponding release markings. Example: ["MAN", "HR"]
"descr"	<STRING> a description of the release group which will be shown in the ClassifyIt classification menu as a 'hint'.

Example:

```
 [{"display":"MAN-CYBER",
  "group":  ["MAN", "CYB"],
  "descr":  "add release markings for MANAGEMENT and CYBER"},
   {"display":"CYBER-DEV-OPS", "group":["CYB", "DEV", "OPS"],
"descr":"Cyber, development and operations"}]
```

(viii) quickSelect

A list of objects which define the entries of the Quick-Select Menu of ClassifyIt. The following attributes are defined per quick-select menu entry:

"display"	<STRING> the entry shown in the quick-select menu. (Note: should be short and easily remembered).
"marking"	<RECORD> defining the classification and release of the quick select entry, using the same "class" and "rel" fields as "classifyit_control"."default_marking" (see Section 5.04(iv)). Two further optional fields: "title" <STRING> shown as the button's ScreenTip, and "descr" <STRING> shown as its SuperTip.

Example:

```
{"display":      "WHITE",
"marking": {
      "class":   "U",
"rel":     []}}
```

(ix) "categories"

(optional, new in 7.1) A list of categories describing the kind of information, for example STAFF or FINANCE. If it is defined, the Release Form shows a CATEGORY column; users can select several categories for every classification but PUBLIC (iCLASS P). With "stanag4774" in "crypto_control", the Release Form also offers ONLY, which makes the selected categories restrictive. iCLASS stores the selected codes as "cat" and ONLY as "only". Each category is defined through the following attributes:

```
"cat"       <STRING> the code stored in iCLASS.
"real"      <STRING> the name shown in the Release Form and in the
            visible marking.
```

Category codes from another organisation's configuration are kept when a marking is changed. A reply or forward in Outlook must keep the categories of the original. Their text in the visible marking is set in "classification_formatting"; whether the Release Form shows the categories is set in "release_form" ("showCategory").

Example:

```
[{"cat": "STAFF", "real": "STAFF"},
 {"cat": "FIN",   "real": "FINANCE"}]
```

(x) "review_control"

(optional, new in 7.1) Review date and downgrade (successor marking) of a document's marking, following STANAG 4774. Without this section the Release Form shows no dates and ClassifyIt does not check them when documents are opened; dates a document carries are kept, and Windows Explorer still shows them. A downgrade is a lower classification, or the same classification released to more nations. The following attributes are defined:

"defaultReviewDays"<INT> the review date offered when the user ticks it for a marking without one, in days from today, e.g. 1825 for five years; 0 = five years (default 0).
"defaultDowngradeDays"<INT> (new in 7.1) the downgrade date offered when the user chooses a lower classification, in days from today, e.g. 1825; 0 = tomorrow (default 0). A downgrade is never added by itself.
"writeReviewDate"<BOOLEAN> the Release Form offers the review date. Ticked, it is recorded in the marking (iCLASS "rev") and in the STANAG 4774 label (drv) and shown in the visible marking; the box shows what iCLASS has. false: no review date is offered or recorded; a changed marking drops one it had (default false; until 7.1 "showReviewDate", still read).
"writeDowngrade"<BOOLEAN> the Release Form offers the downgrade. Once the user chooses the lower classification ("(no downgrade)" = none) it is ticked, recorded in the marking (iCLASS "suc") and in the STANAG 4774 label (dsn, scl) and shown in the visible marking. false: no downgrade is offered or recorded (default false; until 7.1 "showSuccessor", still read).

```
"reviewText"      <STRING> format of the review date in the
                  visible marking; <DATE> is replaced by the date
                  (default "<BR>Review: <DATE>").
"successorText"   <STRING> format of the downgrade; <DATE> is its
                  date, <CLASS> the "real" name of the lower
```

```
                classification (default " - <CLASS> from <DATE>").
    "dateFormat"      <STRING> how dates are shown, e.g. "dd.mm.yyyy"
                (default: the system's short date format).
```

The review date is a reminder only: when it has passed, ClassifyIt asks the user to check the classification when the document is opened, and Windows Explorer shows a ring around the file's stamp. A downgrade does not lower the classification by itself: from its date on, ClassifyIt offers the lower marking when the document is opened. All checks keep using the marking that is applied, so a wrong PC clock cannot lower anything. iCLASS stores the dates as "rev" and "suc" (see iCLASS). Review date and downgrade are written in the format of "formatReleaseMarking".

Example:

```
{"defaultReviewDays": 3650,
 "defaultDowngradeDays": 1825,
 "writeReviewDate": true,
 "writeDowngrade": true,
 "reviewText": "<BR>Review: <DATE>",
 "successorText": "<BR>from <DATE>: <CLASS>",
 "dateFormat": "dd.mm.yyyy"}
```

Texts and format: "reviewText" and "successorText" are the texts of the review date and the successor themselves; the text around them and their font are set in "classification_formatting" (beforeReview ... afterCompletedReview, formatReview). Configurations without those entries keep working: "reviewText" may also contain the label and line breaks, e.g. "
Review: <DATE>".

(xi) "release_form"

(optional, new in 7.1) The parts of the Release Form besides the classification, which is always shown. A part is shown if it is switched on here and has content: the release markings, the categories, and "review_control" - the dates only in the Classify It dialog of a whole document or email. A part that is not shown cannot be changed; what a marking already carries in it is kept. Without this section all parts are shown. Shareware builds show the classification only.

```
    "showRelease"     <BOOLEAN> the Release column, with its filter
                and the release groups (default true).
    "showCategory"    <BOOLEAN> the Category column, with ONLY
                (default true).
    "showReview"      <BOOLEAN> the review date (default true).
    "showDowngrade"   <BOOLEAN> the downgrade (default true).
                "showDates" sets both, as in earlier 7.1 builds.
```

Example - classification and release only:

```
{"showRelease": true,
 "showCategory": false,
 "showReview": false,
 "showDowngrade": false}
```

(xii) "classification_formatting":

A list of attributes which define the formatting of the construction of the classification text which will be inserted by ClassifyIt. The following attributes define the appearance of the

classification. Note that a line break must be inserted through <BR>, do not use \n

"beforeClassification" <STRING> text before the classification.
"beforeClassificationEmail" <STRING> text which is put in front of the classification, but only for Outlook emails.
"beforeClassificationWord" <STRING> text which is put in front of the classification, but only for Word documents.
"beforeClassificationExcel" <STRING> text which is put in front of the classification, but only for Excel sheets.
"beforeClassificationPPT" <STRING> text which is put in front of the classification, but only for PowerPoint slides.
"afterClassification" <STRING> text after the classification.
"beforeAdditionalClassification" <STRING> text before an additional classification, only if the additional marking is not empty.
"afterAdditionalClassification" <STRING> text after an additional classification, only if the additional marking is not empty.
"beforeReleaseStrings" <STRING> text before the release marking, only if the release marking is not empty.
"betweenReleaseStrings" <STRING> text/character between the release markings, only if the release marking is not empty and if more than 2 release markings.
"afterReleaseStrings" <STRING> text after the release marking, only if the release marking is not empty.
"afterCompletedClass" <STRING> text/character after the completed classification and release marking.
"requiresRelAs" <BOOLEAN> (default false) to specify if the "AS" markings defined in the "release_markings" section will be used. (new with version 4.7)
"beforeCategoryStrings" <STRING> (new in 7.1) text before the categories, only if categories are selected.
"betweenCategoryStrings" <STRING> (new in 7.1) text between the categories (default ", ").
"afterCategoryStrings" <STRING> (new in 7.1) text after the categories.
"onlyString" <STRING> (new in 7.1) text after the categories if ONLY is selected (default " ONLY"). Categories are written in the format of "formatReleaseMarking".

"requiresTranslatedLabel" <BOOLEAN> to specify if a translation of the classification and release markings is added to the overall marking.
"beforeTranslatedClassification" <STRING> <see beforeClassification>
"afterTranslatedClassification": <STRING> <see afterClassification>
"beforeTranslatedReleaseStrings" <STRING> <see beforeReleaseStrings>
"betweenTranslatedReleaseStrings" <STRING> <see betweenReleaseStrings>
"afterTranslatedReleaseStrings" <STRING> <see afterReleaseStrings>
"afterTranslatedCompletedClass" <STRING> <see afterCompletedClass>

The following formatting statements define the formatting of the text:

```
"formatBeforeClassification"  list of formatting attributes for
           the classification string
      "font"      <STRING>    defines the font style. E.g.
                  "Arial", "Courier", "Times New Roman"
      "fontSize"  <INT>       defines the size of the font
      "fontColor" <STRING> the font color value in RGB format. A
                  value of "0x000000" specifies to use the value
                  of the classification in the Section
                  5.04(v)"classifications". If black should be
                  specified please use 0x000001
      "fontStyle" list of <STRING> defining the font style, like
                  "bold", "italic", "underline", ...
      "paragraphAlignment"    <STRING> alignment of the paragraph,
                  like "left", "center", "right"

"formatClassification"  list of formatting attributes for the
           release marking. (same as formatBeforeClassification)

"formatReleaseMarking"  list of formatting attributes for the
           release marking. (same as formatBeforeClassification)

"formatTranslation"     list of formatting attributes for the
           release marking. (same as formatBeforeClassification)
```

Example:

```
{"beforeClassification":      "",
 "beforeClassificationEmail": "Classification: ",
 "beforeClassificationWord":  "",
 "beforeClassificationExcel": "",
 "beforeClassificationPPT":   "",
 "afterClassification":       " ",
 "beforeAdditionalClassification":  "<BR>",
 "afterAdditionalClassification":   "<BR>",
 "beforeReleaseStrings":      "REL TO: ",
 "betweenReleaseStrings":     "\\\\",
 "afterReleaseStrings":       "",
 "afterCompletedClass": "<BR>",
 "requiresRelAs": false,

 "requiresTranslatedLabel":              true,
 "beforeTranslatedClassification":  "Einstufung: ",
 "afterTranslatedClassification":   " ",
 "beforeTranslatedReleaseStrings":  "FREIGABE AN: ",
 "betweenTranslatedReleaseStrings": ", ",
 "afterTranslatedReleaseStrings":   "",
 "afterTranslatedCompletedClass":   "<BR>"}

"formatBeforeClassification":
           {"font":    "Courier",
            "fontSize":      12,
            "fontColor":     "0x000000",
            "fontStyle":     ["bold"],
```

```
                "paragraphAlignment":  "left"},

    "formatClassification":
                {"font":    "Courier",
                 "fontSize":      12,
                 "fontColor":     "0x000000",
                 "fontStyle":     ["bold"],
                 "paragraphAlignment":  "left"},

    "formatReleaseMarking":
                {"font":    "Times New Roman",
                 "fontSize":      10,
                 "fontColor":     "0x005500",
                 "fontStyle":     ["italic", "underline"],
                 "paragraphAlignment":  "center"},

    "formatTranslation":
                {"font":    "Arial",
                 "fontSize":      7,
                 "fontColor":     "0x550000",
                 "fontStyle":     ["bold", "italic"],
                 "paragraphAlignment":  "right"}
```

Review and successor: the review date and the successor (downgrade) follow the categories in the marking. Their own texts, with <DATE> and <CLASS>, are "reviewText" and "successorText" of "review_control"; these entries put text around them:

"beforeReview" and "afterReview" <STRING> - before and after the review date text.

"betweenReviewSuccessor" <STRING> - between review date and successor, only when both are shown.

"beforeSuccessor" and "afterSuccessor" <STRING> - before and after the successor text.

"afterCompletedReview" <STRING> - after the whole review/successor part, e.g.
.

Example: "beforeReview": "Review: ", "betweenReviewSuccessor": " - ", "afterCompletedReview": "<BR>" with "reviewText": "<DATE>" and "successorText": "<CLASS> from <DATE>" gives "Review: 2027-11-14 - TLP:GREEN from 2027-11-14" on a line of its own.

Formats: "formatCategory" and "formatReview" - the font of the categories and of the review/successor part, with the same entries as "formatClassification" (font, fontSize, fontColor, fontStyle, paragraphAlignment). Without them, both are written like the release ("formatReleaseMarking"); a missing entry inside a block keeps the release's value.

(xiii) "x_header_config"

This section is no longer read by ClassifyIt 7.0 and can be omitted from the configuration file. The marking property name is now configured via "propertyKeyName" in the "classifyit_control" section (see Section 5.04(iv)), and release codes are stored as a JSON array under "rel" rather than a delimiter-joined string, so "xReleaseCodeDelimiter" no longer applies either.

"xMarkingName" (obsolete) formerly the name of the marking property/X-Header; replaced by "classifyit_control"."propertyKeyName".
"xReleaseCodeDelimiter" (obsolete) release codes are now stored as a JSON array, so no delimiter is needed.

Example:

```
{"xMarkingName":              "Classification",
"xReleaseCodeDelimiter":      "-"}
```

(xiv) "outlook_control"

A list of attributes which define specific Outlook application behaviour. The following attributes are defined:

"banned_filetypes" [<STRINGS>] (list of strings) defining filetypes which the user is not allowed to send by email. IMPORTANT is to specify the preceding '.' as well.

"allowLowerClassification" <BOOLEAN> specifies if a forward or reply to an email could have a lower classification than the original email.

"warnLowerClassification" <BOOLEAN> specifies if, for a forward or reply to an email with a lower classification than the original email, the user should get a warning message. This setting is only effective if "allowLowerClassification" is true.

"allowLowerAttachmentClassification" <BOOLEAN> controls whether the email classification may be lower than the classification of an inspected attachment. If false, sending is blocked when an inspected attachment has a higher classification or has no valid marking. The check covers supported Word, Excel and PowerPoint files, and PDFs carrying the ClassifyIt marking; it does not inspect the contents of every attachment format or a TDF payload.

"warnLowerAttachmentClassification" <BOOLEAN> displays a confirmation warning when allowLowerAttachmentClassification is true and an inspected attachment has a higher classification than the email or has no valid marking. The user can cancel sending or explicitly continue.

"automaticallyEncryptAllAttachments" <BOOLEAN> applies Generic protection to attachments that are not already .tdf or .cit files. Recipients need ClassifyIt to decrypt Generic files, but decryption does not require a licensed build. Cancelling or failing attachment processing blocks the send attempt. The extension check does not establish that an existing container is encrypted. In shareware builds, exports are unencrypted, signed CI-TDF containers regardless of the selected mode.

"markSubject" <BOOLEAN> specifies if the classification should be added in the Subject line of the email. If true the classification will be shown within [ ] brackets.

"markSubjectWithRealName" <BOOLEAN> specifies if the subject marking uses the real name of the classification or the xClassification.

"setSensitivityLevel" <BOOLEAN> specifies if ClassifyIt will change the Email Sensitivity field.

Example:

```
{"banned_filetypes":    [".dll", ".exe", ".bat"],
 "allowLowerClassification": true,
 "warnLowerClassification": true,
 "allowLowerAttachmentClassification": false,
```

```
 "warnLowerAttachmentClassification":  true,
 "automaticallyEncryptAllAttachments": false,
"markSubject":                          false,
 "markSubjectWithRealName":         true,
 "setSensitivityLevel":             false}
```

Emails started with Send to > Mail recipient in Windows Explorer (Simple MAPI) open in plain-text format, in which the protected area for the classification marking cannot be created. ClassifyIt writes the marking as normal text instead. It is verified when the email is sent; if it was changed, sending is stopped and the classification is applied again. Switching such an email to HTML or Rich Text and applying the classification again gives the protected marking.

(xv) "word_document_control"

A list of attributes which define for the Microsoft Word application where the classifications should be added and if those are to be enforced. The following attributes are defined:

"addToHeader" <BOOLEAN> if true the classification will be added to the header of the document.

"addToFooter": <BOOLEAN> if true the classification will be added to the footer of the document.

"enforceMarking": <BOOLEAN> specifies if the user can save or print Word documents without a security marking. If set to true the document must have a security marking.

"paragraphMarking": <BOOLEAN> specifies if the user can mark individual paragraphs in the Word document. THIS FEATURE IS IN BETA STATUS AND SHOULD ONLY BE USED IN TEST ENVIRONMENTS. For production this value shall be false

Example:

```
{"addToHeader":         true,
 "addToFooter":         true,
 "enforceMarking":      false,
 "paragraphMarking":    false}
```

(xvi) "excel_control":

A list of attributes which define for the Microsoft Excel application where the classifications should be added. The following attributes are defined:

"addToHeader" <BOOLEAN> if true the classification will be added to the header of the document.

"addToFooter" <BOOLEAN> if true the classification will be added to the footer of the document.

"enforceMarking" <BOOLEAN> specifies if the user can save or print Excel sheets without a security marking.

Example:

```
{"addToHeader":         true,
 "addToFooter":         true,
 "enforceMarking":      true}
```

(xvii) "powerPoint_control":

A list of attributes which define for the Microsoft PowerPoint application where the classifications should be added. The following attributes are defined:

```
"enforceMarking"        <BOOLEAN> specifies if the user can save
            or print PowerPoint slides without a security marking.
"posLeft"   <INTEGER> specifies the left position of the
            classification text on the slide.
"posTop"    <INTEGER> specifies the top position of the
            classification text on the slide.
```

Example:

```
{"enforceMarking":      true,
 "posLeft":             50,
 "posTop":              200}
```

(xviii) "crypto_control":

A list of attributes which define if the CryptIt functionality is available, allowing the export of documents to encrypted Trusted Data Format (TDF) containers, and if email attachments could be encrypted to TDF containers. By default this should be enabled:

```
"allowEncryption" <BOOLEAN> if true CryptIt is enabled.
"stanag4774"      <BOOLEAN>(optional, new in 7.1) if true every
            CI-TDF container also carries a STANAG 4774
            confidentiality label (see CI-TDF Containers).
"stanag4774Policy"      <STRING>(optional, new in 7.1) the policy
            identifier of the label ("pol"), for example DEU or
            NATO; default "originator" (in "classifyit_control").
"stanag4774Context"     <STRING>(optional, new in 7.1) for the
            policy NATO only: the value of the mandatory Context
            category (ADatP-4774, Appendix 2), the NATO context
            with its current suffix as the NATO Security Policy
            defines it. Without it no label is written for NATO.
```

New in ClassifyIt 7.0: encryption to a CI-TDF container is also available directly from Word, PowerPoint and Excel documents (not only Outlook email attachments), via the 'Crypt It' ribbon menu's 'Export' item. The same 'allowEncryption' setting above also controls whether this Word/PowerPoint/Excel Crypt It menu is shown.

Selecting 'Export' prompts the user for one of four encryption modes and creates a CI-TDF container (named <original file name>.tdf) alongside the source document:

'None' - exports the classified document unencrypted (only signed and tagged with its classification).

'Generic' - uses the preshared certificate built into ClassifyIt. Any ClassifyIt installation can decrypt the content, including shareware builds; this mode is not recipient-specific.

'PIN' - encrypts with a user-supplied PIN (at least 8 characters is recommended); the PIN must be communicated to the recipient out-of-band.

'Certificate' - encrypts to the selected recipients' registered public-key certificates and automatically includes the sender's own certificate, so each included private-key holder can decrypt. The recipients are selected in a list in which each click selects or deselects a certificate, so several recipients can be chosen.

Document properties (Title, Subject, Category, Keywords, Comments, Author, etc.) are stored as JSON in the separate 0.metadata entry of the CI-TDF ZIP container. The signed manifest contains the metadata hash, which is verified during import. Metadata is stored in clear text even when the document payload is encrypted; include only properties appropriate for disclosure.

The same menu's 'Import' item decrypts a .tdf container and re-opens the original document, restoring its classification from the verified container manifest rather than from the file's own internal properties.

The menu's Certificate Management entry opens the common window for generating a certificate/key pair, registering an existing identity, viewing imported public certificates and checking the current certificate. Certificates and private keys are stored in %APPDATA%\Certificates. Outlook's Attach My Public Key command sends a signed, unencrypted .crt.tdf package and may request the sender's private-key password; the recipient imports it with Save a Public Key.

CI-TDF processing limits: the stored payload is limited to 256 MiB and the restored document to 512 MiB. Manifest JSON is limited to 4 MiB and metadata to 16 MiB. PIN imports accept PBKDF2 counts from 100,000 to 2,000,000 iterations; current exports use 600,000. Files exceeding these limits are rejected. Reduce or split oversized source files before exporting.

Upgrade and recovery: ClassifyIt 7.0 does not decrypt legacy .cit files. Before upgrading, retain a compatible older ClassifyIt environment and its required keys, decrypt legacy files there, and re-export them to CI-TDF using v7. Test recovery on representative files and retain the originals until migration is verified.

Shareware builds create signed, unencrypted CI-TDF containers even if an encryption mode was selected, and display a notice for explicit export/encryption actions. Decryption remains available independently of licensing. Verify the deployed build and licensing before enabling a policy that requires encryption.

Example:

```
{"allowEncryption":          true,
 "stanag4774":false}
```

STANAG 4774 in documents: with "stanag4774" switched on, Word, Excel and PowerPoint documents also carry the STANAG 4774 label (ADatP-4774.7 JSON) of their marking: as a custom XML part of the namespace urn:ugarbe:classifyit:stanag4774:1, written before every save so it always matches the saved marking, and removed when the option is off or the document has no marking. PDFs exported from Office carry it as the metadata entry STANAG4774. Show Assertions in Explorer displays it; Clear Classification removes it.

Article VI. ClassifyItUserData.json

This file is located in the users' roaming Application Data directory (CSIDL_APPDATA, typically: C:\Documents and Settings\username\Application Data\Roaming).

If not existing, the file is created by ClassifyIt during its startup. The file is edited by ClassifyIt when the user adds or removes preferences to the content of some functions. The file can be edited, however it is important that a JSON compliant file is created/maintained, otherwise ClassifyIt will not work properly.

The following sections are defined in the ClassifyItUserData.json file:

In Word Application – ClassifyIt Property Menu:

Section 6.01 "categories"

```
"categories"      An array of strings which can be selected for
            adding categories to the CATEGORIES_PROPERTY
Example: "categories": ["CONTRACT", "INVOICE", "REPORT"]
```

Section 6.02 "keywords"

```
"keywords"  An array of strings which can be selected for adding
            keywords to the KEYWORDS_PROPERTY
Example: "keywords": ["TEMPLATE", "DOCUMENT"]
```

Section 6.03 "templates"

```
"templates" An array of records which the user can use to add
            multiple properties to all property fields. Each
            template has a unique name.
Example: "templates": [
            {"templateName":"Temp1",
             "subject":       "Subject",
             "title":         "Title",
             "keywords":      "Keyword1, Keyword2",
             "category":      "Cat1, Cat2",
             "comments":      "Comment",
             "lastauthor":    "Last Author",
             "manager":       "Manager",
             "format":        "Format",
             "company":       "Company",
             "hyperlinkbase": "LinkBase",
              "author":"Author"}]
```

Section 6.04 "ECC"

```
"ECC" is an object whose "myCert" field identifies the current
            user certificate by its base name.
            The %APPDATA%\Certificates folder contains the user
            certificate and private key, together with imported
            recipient public certificates. Back up private keys
            and retain the keys needed to decrypt older files when
            replacing an identity.
```

```
Example: {"ECC": {"myCert": "bob@ugarbe.de"}}
```
mailto:bob@ugarbe.de

Section 6.05 "quickSelect"

```
"quickSelect" is an array of user Quick Select entries. Each entry
            has a "display" string and a "marking" object
            containing "class" and a "rel" array. The codes must
            match the configured classification and release codes.
            An optional "icon" string selects the stamp icon. Key
            names are case-sensitive; do not use the obsolete
            "classification", "release" or delimiter-separated
            release format.
Example: {"quickSelect": [
  {"display":"Open rel Management/Finance",
   "marking":{"class":"O", "rel":["MAN", "FIN"]}
  }]}
```

Article VII. Manual Installation

ClassifyIt can be installed without the setup program - by hand, or by the scripts of a software distribution (see Deployment Phase). The procedure is shown for the 64-bit version. The 32-bit version is equivalent, with the 32-bit regsvr32 (%SystemRoot%\SysWOW64\regsvr32.exe), the folder %ProgramFiles(x86)%\ClassifyIT, the registry key HKLM\SOFTWARE\WOW6432Node\CLASSIFY_IT and /reg:32 instead of /reg:64, and the 32-bit files of the same build. The add-in must match the bitness of Office.

Section 7.01 Files

All files must come from the same build:

ClassifyIt64.dll - the Office add-in (64-bit Office); with 32-bit Office ClassifyIt32.dll, and ClassifyIt64.dll for ClassifyItTool and Explorer, which are always 64-bit

Until 7.1.0.0 the add-in was ClassifyIt.dll. The 7.1.0.1 setup deletes it and registers ClassifyIt64.dll (or ClassifyIt32.dll) in its place - the same add-in IDs, so Office needs no other change. For 32-bit Office use the setup "…(32-bit-Office)…"; it checks the bitness of the installed Office (/SKIPOFFICECHECK skips it).

libcrypto-3-x64.dll (32-bit Office also libcrypto-3.dll) - OpenSSL, loaded only from this folder; libssl-3-x64.dll is installed but not used

ClassifyIt.json - the configuration file (or its locations on a network share or web server)

ClassifyItTool.exe - the Windows Explorer integration and Certificate Management (optional)

ClassifyItShell.dll - the classification stamp on file icons (optional, with ClassifyItTool.exe)

Section 7.02 Procedure

Close Word, Excel, PowerPoint and Outlook first. Run the commands in a Command Prompt as administrator - the examples use the folder C:\Program Files\ClassifyIT.

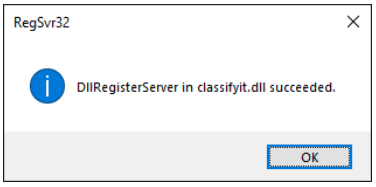
1. Copy the files into the program folder, for example C:\Program Files\ClassifyIT. Users must not be able to change files in that folder.

2. Register the add-in for Word, Excel, PowerPoint and Outlook, for all users of the computer:

```
"%SystemRoot%\System32\regsvr32.exe" /s "C:\Program Files\ClassifyIT\ClassifyIt64.dll"
```

This registers the COM classes and the add-ins ClassifyIt.WordAddIn, ClassifyIt.ExcelAddIn, ClassifyIt.PowerPointAddIn and ClassifyIt.OutlookAddIn under HKLM\Software\Microsoft\Office\<application>\Addins (LoadBehavior 3:

loaded at start). Without /s, regsvr32 confirms the registration:



3. Link the configuration file (see Registry Keys) - a file path, a network share or a web address, for example C:\Program Files\ClassifyIT\ClassifyIt.json; the second location is used when the first cannot be reached:

```
reg add "HKLM\SOFTWARE\CLASSIFY_IT" /v ConfigFile /t REG_SZ /d
"<LOCATION_1>" /f /reg:64
reg add "HKLM\SOFTWARE\CLASSIFY_IT" /v AlternateConfigFile /t REG_SZ /d
"<LOCATION_2>" /f /reg:64
```

4. Register the event log (only with "logEvents"): the log ClassifyIt and its source, which takes the descriptions of its events from ClassifyIt64.dll (see Event Logging):

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt" /v
CustomSD /t REG_SZ /d
"O:BAG:SYD:(A;;0xf0007;;;SY)(A;;0x7;;;BA)(A;;0x2;;;IU)" /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt" /v
MaxSize /t REG_DWORD /d 33554432 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt" /v
Retention /t REG_DWORD /d 0 /f
reg add
"HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt\ClassifyIt"
/v EventMessageFile /t REG_EXPAND_SZ /d "C:\Program
Files\ClassifyIT\ClassifyIt64.dll" /f
reg add
"HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt\ClassifyIt"
/v TypesSupported /t REG_DWORD /d 6 /f
```

Only administrators and SYSTEM can read the log; signed-in users can only write to it. If the program folder changes later, change EventMessageFile too.

5. Set up the Explorer integration for every user (optional, with ClassifyItTool.exe): Windows Active Setup runs ClassifyItTool.exe /register once for each user at their next sign-in:

```
reg add "HKLM\SOFTWARE\Microsoft\Active Setup\Installed
Components\{14911B57-A4ED-4B56-92B6-87F636B4A5F8}" /ve /d "ClassifyIt
Explorer integration" /f /reg:64
reg add "HKLM\SOFTWARE\Microsoft\Active Setup\Installed
Components\{14911B57-A4ED-4B56-92B6-87F636B4A5F8}" /v StubPath /d
"\"C:\Program Files\ClassifyIT\ClassifyItTool.exe\" /register" /f /reg:64
reg add "HKLM\SOFTWARE\Microsoft\Active Setup\Installed
Components\{14911B57-A4ED-4B56-92B6-87F636B4A5F8}" /v Version /d "7,1,0,0"
/f /reg:64
reg add "HKLM\SOFTWARE\Microsoft\Active Setup\Installed
Components\{14911B57-A4ED-4B56-92B6-87F636B4A5F8}" /v IsInstalled /t
REG_DWORD /d 1 /f /reg:64
```

For a user who is signed in already, run "C:\Program Files\ClassifyIT\ClassifyItTool.exe" /register as that user - it needs no administrator rights. Raise "Version" with every update, so that it runs again for every user.

6. Prevent disabling (recommended): set the registry values described in Prevent Disabling of ClassifyIt.

7. Check: start Word - the ClassifyIt tab is shown, and File > Options > Add-ins lists ClassifyIt under the active application add-ins. With the Explorer integration, a right-click on a file shows the ClassifyIt menu.

Section 7.03 Explorer Integration

ClassifyItTool.exe /register sets up the Explorer integration for the user who runs it: the Export as CI-TDF... command, opening .tdf files, and the classification stamp on the icons of .tdf, Office and PDF files. ClassifyItTool.exe /unregister removes it and restores the previous file-type settings exactly. Neither needs administrator rights. ClassifyItTool.exe /unregister /allusers, run as administrator, removes it for every user of the computer - the uninstaller does this.

ClassifyIt reports results as Windows notifications (bottom right, kept in the notification centre). Windows shows them under the identity "ugarbe.ClassifyIt", which ClassifyIt registers for each user on its first notification (HKEY_CURRENT_USER\Software\Classes\AppUserModelId\ugarbe.ClassifyIt, with the name ClassifyIt and its icon) - no Start menu shortcut is needed, also not for a manual installation. ClassifyItTool /unregister removes the registration. If notifications are switched off for ClassifyIt in the Windows settings, important ones appear as a dialog instead. Errors that need an action are always dialogs.

The integration is stored per user under HKEY_CURRENT_USER\Software\Classes; what ClassifyIt changed is recorded under HKEY_CURRENT_USER\Software\ClassifyIt, so that /unregister can restore it. For Word, Excel and PowerPoint files ClassifyIt registers its own preview handler (ClassifyItShell.dll, CLSID {486BFBEF-D49B-4A42-BAA3-471FCBE57274}): it shows the preview picture stored in the file, with the classification stamp. PDF previews come from the PDF program and carry no stamp.

For mass deployment, Windows Active Setup runs /register for every user (step 5 above, and Deployment Phase).

ClassifyItTool.exe restartshell (also /restartshell) restarts Windows Explorer for the signed-in user: open folder windows are remembered and reopened; Explorer and the preview helpers running ClassifyItShell.dll are ended in this session only; the icon and preview caches are deleted. Each run writes %TEMP%\ClassifyIt-restartshell.log. It does not run elevated while Explorer runs normally. /unregister restarts Explorer the same way. /register does not restart it (Active Setup and the setup call it), but deletes the caches when Explorer is not running.

The Active Setup version is 7,1,0,1: users who already had 7.1.0.0 run /register again at their next sign-in, which registers the preview handler for them.

Article VIII. Manual Deinstallation

The procedure is shown for the 64-bit version; the 32-bit version is equivalent (see Manual Installation). Run the commands in a Command Prompt as administrator - the examples use the folder C:\Program Files\ClassifyIT.

1. Stop ClassifyIt. Ask the users to save and close Word, Excel, PowerPoint and Outlook, and the documents they opened from CI-TDF containers. Then stop the ClassifyIt background tool - it asks about container documents that are still open, and saves their last changes into the containers:

```
"C:\Program Files\ClassifyIT\ClassifyItTool.exe" /quit
```

If it still runs afterwards, for example in the session of another user:

```
taskkill /f /im ClassifyItTool.exe
```

2. Remove the Explorer integration for every user of the computer - this also restores their previous file-type settings and the Office previews:

```
"C:\Program Files\ClassifyIT\ClassifyItTool.exe" /unregister /allusers
```

3. Deregister the add-in:

```
"%SystemRoot%\System32\regsvr32.exe" /u /s "C:\Program Files\ClassifyIT\ClassifyIt64.dll"
```

4. Remove the registry keys of the configuration file and of Active Setup:

```
reg delete "HKLM\SOFTWARE\CLASSIFY_IT" /f /reg:64
reg delete "HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{14911B57-A4ED-4B56-92B6-87F636B4A5F8}" /f /reg:64
```

The values set for Prevent Disabling of ClassifyIt are removed the way they were set (registry or Group Policy).

The event log is kept, with its events. To remove it too - this deletes the recorded events - export them first if they are still needed:

```
wevtutil epl ClassifyIt "C:\ClassifyIt-events.evtx"
reg delete "HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt" /f
```

The log file %SystemRoot%\System32\winevt\Logs\ClassifyIt.evtx can be deleted after Windows has been restarted.

5. Delete the program folder:

```
rmdir /s /q "C:\Program Files\ClassifyIT"
```

Windows Explorer keeps ClassifyItShell.dll loaded in the sessions of signed-in users. If the file cannot be deleted, restart Windows Explorer, or delete the folder after the users have signed out or after a restart.

6. Remove the user data (optional) - in every user profile:

%APPDATA%\ClassifyItUserData.json - the user's own settings: categories, keywords, templates, Quick Select

%APPDATA%\ClassifyIt.json and %APPDATA%\ClassifyIt.log - the local copy of the configuration file and the log

%APPDATA%\Certificates - the user's own certificate and private key, and the trusted certificates. Delete it only if the user no longer needs them: without the private key, containers encrypted for the user cannot be opened any more.

%LOCALAPPDATA%\ClassifyIt - the icon of the Windows notifications

%LOCALAPPDATA%\cli_temp_import - the decrypted working copies of CI-TDF containers. Before deleting it, make sure that no working copy holds changes that are not in its container: ClassifyIt offers them when Word, Excel or PowerPoint or ClassifyItTool starts (see Saving Back).

The registry entries of the Explorer integration (HKEY_CURRENT_USER\Software\ClassifyIt and the file types under HKEY_CURRENT_USER\Software\Classes) are removed by step 2, the notification identity (HKEY_CURRENT_USER\Software\Classes\AppUserModelId\ugarbe.ClassifyIt) as well.

Article IX. Classification Properties (X-Header Information)

ClassifyIt adds Properties and X-Header information to documents and emails.

The properties and X-Header information are Name: Value pairs. The name is specified in the configuration file under the "classifyit_control" section by "propertyKeyName" (see Section 5.04(iv)). The value is a JSON string with the following structure:

```
{"org": "ACME",
 "class": "U",
 "rel":   []}
```

"org" identifies the originating organisation (configured via "originator" in "classifyit_control"). "class" and "rel" can be used by Data Leak Prevention (DLP) - their values are those of the "class" and "rel" fields in the "classifications" and "release_markings" sections, representing the actual classification and release assigned to the document.

We recommend using a standardised marking such as iCLASS, so that different deployments can recognise classification markings of each other.

Article X. Data Leak Protection (DLP)

Every document classified by ClassifyIt will, in addition to the human readable classifications, be tagged with machine readable property data. The Classification Properties section describes the data structure of the property data.

XML Guards or Application Guards can read the classification and release values to determine if the email or document can be transferred.

Article XI. Prevent Disabling of ClassifyIt

Section 11.01 Prevent ClassifyIt being disabled by a user

In general, Office add-ins can be disabled by a user. To prevent this for Office 2010, the following registry keys need to be created:

Outlook:

Key: HKEY_CURRENT_USER\Software\Microsoft\Office\14.0[1]\Outlook\Security

Value: DontDisplayHKLMAddins (REG_DWORD) set value to 1

[1] 16.0 for Office 2019 and Office 365

Section 11.02 Ensure ClassifyIt is loaded at all times

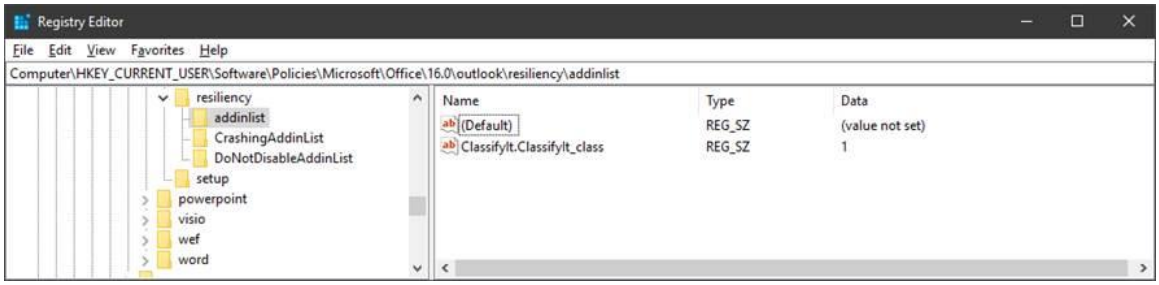
Office evaluates the health and loading time of add-ins. ClassifyIt is sometimes flagged as taking too long to load. To prevent this set the following registry keys:

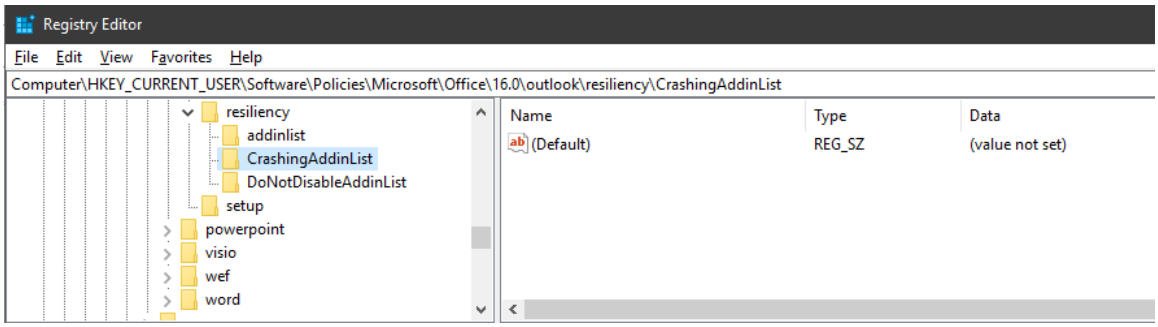
<ClassifyIt ID>: ClassifyIt.ClassifyIt_class

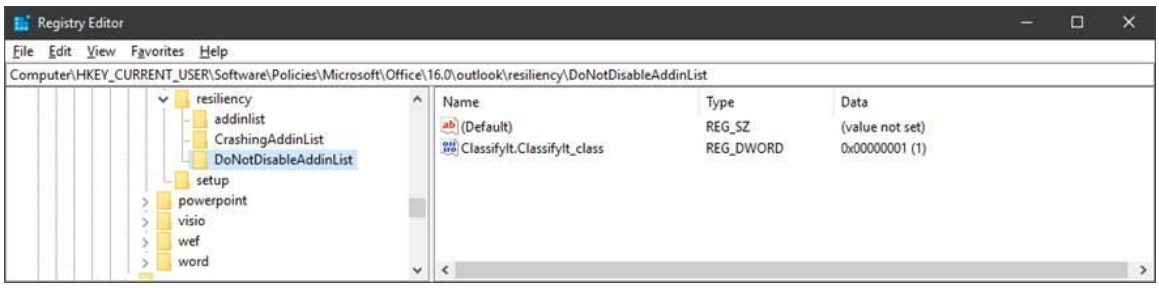
Go to registry key:
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\16.0\Outlook\Resiliency

Ensure the <ClassifyIt ID> is not listed in:
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\16.0\Outlook\Resiliency\CrashingAddinList

Add the <ClassifyIt ID> to:
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Office\16.0\Outlook\Resiliency\DoNotDisableAddinList as a DWORD with the value 1 (see below):







Article XII. Event Logging

With "logEvents" set to true in "classifyit_control", ClassifyIt records what users do with classified documents and emails:

Read - a classified document, CI-TDF container or email is opened

Write - a document is saved, saved under a new name, or exported to PDF or CI-TDF

Send - an email is sent in Outlook, or ClassifyIt blocked it

Each record holds the time, user, computer, application and the iCLASS marking, and - depending on the action - the path and the type of destination, the previous marking, or the subject, recipients and attachment names of the email. The contents of documents, emails and attachments are never recorded. Paths, subjects and addresses are sensitive information themselves: protect the records accordingly.

The records are written to the Windows event log. If an event cannot be written, the user's action still goes ahead; it is never blocked.

Section 12.01 Windows Event Log

The setup registers the log ClassifyIt (Event Viewer > Applications and Services Logs > ClassifyIt) with the source ClassifyIt; step 4 of Manual Installation does the same by hand. Each event shows its values as labelled lines: the application, the action, the marking as it is shown in the document and as iCLASS, and the path or the subject, recipients and attachments of the email. The time, user and computer are those of the event itself. For a SIEM, every value is a field of its own (EventData):

1001 Read - Information

1002 Write - Information

1003 Send allowed - Information. ClassifyIt let the email go; this does not confirm that it was delivered.

1004 Send blocked - Warning, with the reason. An ID of its own, so that alerts can be raised on it.

Only administrators and SYSTEM can read or clear the log - open Event Viewer as administrator. Signed-in users can only write to it. The log holds 32 MB and overwrites its oldest events when full. To let another account read the log, for example the collector of a SIEM, add it to the value CustomSD of HKLM\SYSTEM\CurrentControlSet\Services\EventLog\ClassifyIt (read access is 0x1); the group Event Log Readers has no access. Upgrades keep changes to CustomSD, MaxSize and Retention.

Without this registration nothing is recorded; ClassifyIt never writes to the Application log. Deinstalling ClassifyIt keeps the log and its events (see Manual Deinstallation).

Section 12.02 Limits

A save is recorded when Office has written the file; cancelled and failed saves are not recorded. A document closed before ClassifyIt could see its save - for example when it is saved from the close prompt - is recorded if its file changed; a new name chosen while closing is not known then, so the old name is recorded. Documents in a hidden Office application, and new documents that were never saved, are not recorded as read. A document is recorded as read once each time it is opened. A value longer than 3,000 characters, such as a very long list of recipients, is cut and ends with [truncated]; the event is still written.

Article XIII. Information Manager

Input from the information manager (office) is important so that ClassifyIt is properly configured to reflect your needs for information security and classification.

The figures below provide an idea of what should be defined by the information manager. The classifications your company/administration is using to classify documents and emails, release markings, if those are used, and if so, wider release groups could be used.

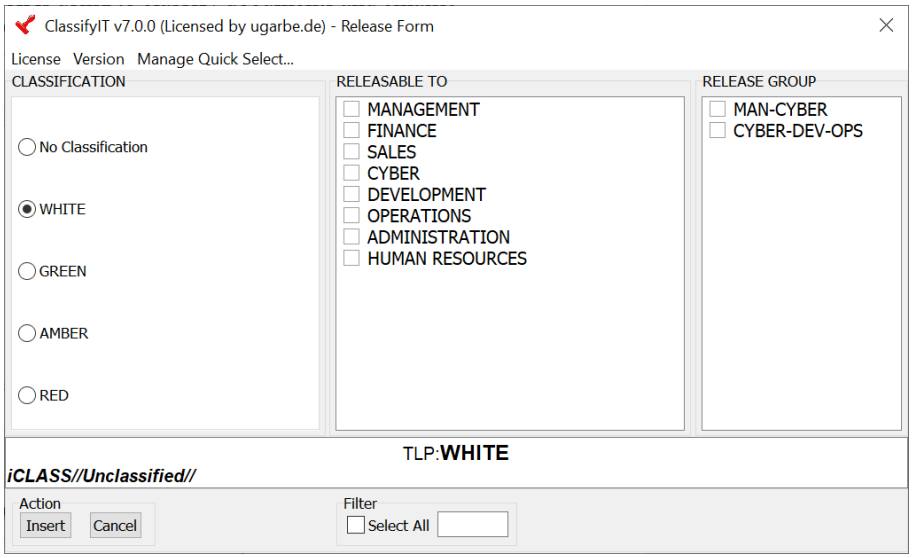


Figure 1 – Traffic Light Protocol Classification Marking – White, no release

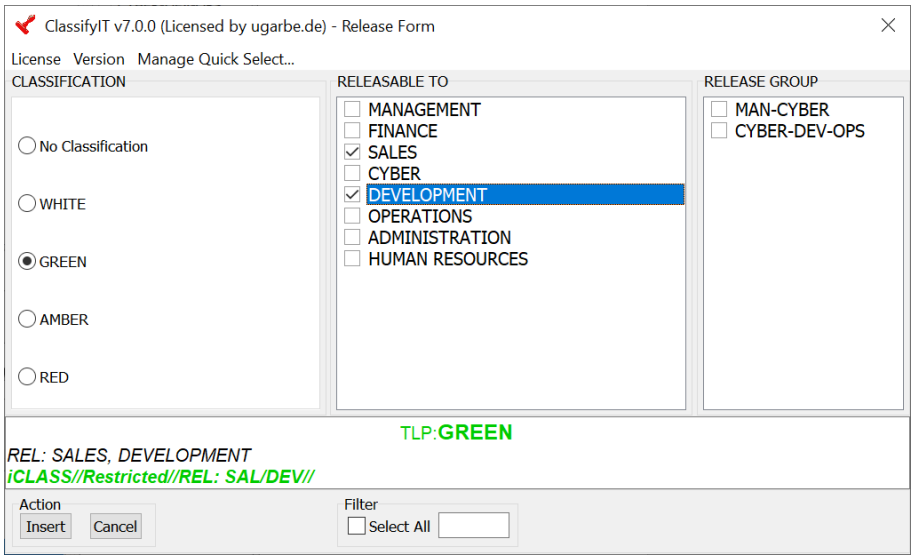


Figure 2 - TLP Marking - Green, with release

The figures above show the ClassifyIt Release Form with an example of the Traffic Light Protocol (TLP) markings (White, Green, Amber, Red). Other markings might be: Unclassified, Restricted, Confidential, Secret, Top Secret. ClassifyIt provides you with

the option of any number of markings in any language and any Unicode character-set (UTF-8). ClassifyIt shows with the selection of classification and release what the actual marking looks like which could be inserted in the document the user is working on (below the selection areas).

The examples above also show a feature of ClassifyIt to provide a 'translation' of the markings. This is an important feature for information exchange between different companies or organisations, when not using the same classification marking. ugarbe.de suggests the use of an International Classification (iCLASS) which is well structured using standard classification markings in English language.

https://classifyit.eu/case-study/iclass

Section 13.01 Classifications

List all classifications used by your organisation, in the order of lowest (e.g. public) level to the highest most protective level (e.g. secret). Optional, you could provide a screen tip the user would see when selecting a classification.

<please override the TLP example below with your policy>

1) White
    a) Screen Tip: Public Information, can be shared with everyone
    b) Translation: Unclassified
2) Green
    a) Screen Tip: Information can be shared with Peers and Partner Organisations
    b) Translation: Restricted
3) Amber
    a) Screen Tip: Information is critical and must be handled with care
    b) Translation: Confidential
4) Red
    a) Screen Tip: Information is very critical and if wrongly distributed would have grave damage to our organisation
    b) Translation: Secret

Section 13.02 Release Markings

List all of the release markings your organisation uses, with a screen tip for the user. This is optional. If no release markings are necessary, please provide an empty list.

<please override the example below with your policy>

1) Management
    a) Screen Tip: Members of Management
    b) Translation:
2) Finance
    a) Screen Tip: Office of Finance
    b) Translation:
3) …
4) COMPANY_1
    a) Screen Tip: Company 1
    b) Translation:
5) COMPANY_2
    a) Screen Tip:
    b) Translation:
6) COMPANY_3
    a) Screen Tip:
    b) Translation:
7) …

Section 13.03 Release Groups

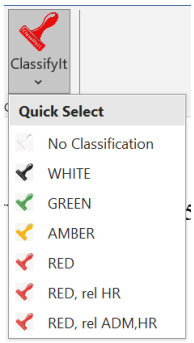
List the release groups which might be useful in your organisation. For a Release Group think about a named project your company has with other companies – for this named project you could define the project name as a Release Group, which would include the companies (e.g. COMPANY_1, COMPANY_3, COMPANY_8) – this makes it easier for the user to select the correct companies for sharing information. This feature is optional and only makes sense if you use Release Markings.

<please override the example below with your policy>

1) PROJECT_1
    a) Members: COMPANY_1, COMPANY_3, COMPANY_8
    b) Screen Tip: Project 1 is about xxx

2) PROJECT_2
    a) Members: COMPANY_1, COMPANY_2, COMPANY_5, COMPANY_6
    b) Screen Tip: Project 2 is about xxx
3) …

Section 13.04 Quick Select



Provide your users a short list of frequently used classification markings. The marking has a display name and defines a classification and release marking – the release marking could be empty.

It is recommended to provide at least the classifications for this feature.

Users can modify and adjust this list later on while using ClassifyIt. However, it is recommended to provide a starting list of frequently used markings.

ClassifyIt restricts the number of Quick Select entries to 10.

<please override the example below with your policy>

1)
    a) Display: WHITE
    b) Classification: White
    c) Release:
2)
    a) Display: GREEN
    b) Classification: Green
    c) Release:
3) …
4)
    a) Display: G rel M+F
    b) Classification: Green
    c) Release: Management, Finance
5) …

Section 13.05 Default Marking

Select which default marking should be applied to a new document or email. Normally this is either the lowest classification without a release, or the highest classification without a release. However, any defined classification and release could form the default marking.

<please override the example below with your policy>

1) Classification: White
2) Release: <>

Section 13.06 Formatting

Last but not least the formatting of the classification marking needs to be described. ClassifyIt provides extensive options to display classification markings in the way as required by your policy.

To best approach the definition of the formatting, consider some examples. Consider the following:

- A classification in Outlook, Word, PowerPoint and Excel could have different preceding text. This preceding can have its own colour, font size, etc.
    - Example: some customers prefer for emails the expression "Classification:" in front of the classification.
    - Example: the classification of this document (WHITE) is preceded with the expression "TLP:" to indicate this is a TLP marking.
    - Preceding Outlook: < please fill >
    - Preceding Word: < please fill >
    - Preceding PowerPoint: < please fill >
    - Preceding Excel: < please fill >
    - Preceding Formatting:
-

Article XIV. CI-TDF Containers

Section 14.01 Signatures and Trust

Every container is signed with the creator's own certificate, in all modes. Without a certificate, PIN containers are signed with a key derived from their content key, None and Generic containers with the ClassifyIt Generic key.

A certificate is trusted when it is in the user's trusted certificates (%APPDATA%\Certificates). For a container signed by any other certificate, ClassifyIt shows the signer and its fingerprint - as hexadecimal groups, as nine words and as a safety number - and asks the user to confirm the sender before the container is opened.

Section 14.02 Saving Back

A document opened from a container is written back into it when it is saved: by the add-in for Word, Excel and PowerPoint, by ClassifyItTool for all other file types. Any organisation writes it back; the warning names the organisation the container comes from. The container is overwritten after a warning at every save, PIN containers only with their PIN; Certificate containers for the same recipients. The container gets the document's marking; changing it is the user's responsibility. Save as new container names the file in the manifest after the new container. The container keeps its identifier, and its classification date while the marking is unchanged. If a save-back does not happen, the user chooses Retry, Save as new container, Keep for later or Discard changes. A container that someone else has changed since the document was opened is overwritten only if the user confirms (No is the default).

The manifest carries exactly the document's marking: originator, classification, release, categories and ONLY, review date, downgrade and the classification date. Opening a container gives the working copy this marking, so the review and downgrade checks work for containers too.

Section 14.03 STANAG 4774 Confidentiality Label

With "stanag4774" set to true in "crypto_control", each container's manifest also carries its classification as a STANAG 4774 confidentiality label in the JSON syntax of ADatP-4774.7, bound to the content by a STANAG 4778 Binding Data Object in the JSON syntax of ADatP-4778.7 (a JWS signed with the creator's certificate). It is meant for systems that read NATO labels; ClassifyIt itself decides on iCLASS only.

The label is taken from the configuration as it is, without a separate mapping: the policy is "stanag4774Policy", else "originator" (in "classifyit_control"), the classification name is "real" of the classification, and the release is a permissive category named after "beforeReleaseStrings" without its formatting (for example "
FREIGABE: " gives FREIGABE), holding the release codes. Categories are added by their "real" names - with ONLY as the restrictive category Additional Sensitivity, without as the informative category Administrative. STANAG 4774 requires every category to carry its name; ClassifyIt uses the names of ADatP-4774 (Table 11) in every policy. The review date

becomes the label's review date (drv), a downgrade its succession date (dsn) with the successor's label (scl). The creation date (dcr) is, in a document, the time the label was first assigned (UTC) - kept while the label stays the same, renewed when it changes; in a CI-TDF container it is the classification date. Without a downgrade the label always has a review date, as STANAG 4774 requires: a marking without one gets the classification date plus "defaultReviewDays" (five years if it is 0). A classification whose "real" is empty gets no label. The level is the same in every policy through iCLASS, so no equivalence tables are needed.

In Word, Excel and PowerPoint files the label can be bound and signed (STANAG 4778) in the JSON form of the CI-TDF container: the custom XML part then holds the same handling assertion as a container manifest - the label as its statement, and a Binding Data Object (ADatP-4778.7) whose JWS carries the label and a reference to every part of the document (pack URI and SHA digest), signed with the user's certificate; profile ClassifyIt-OOXML-1.0, scope "document". The parts are those ADatP-4778.2 (Chapter 4, Tables 12 and 13) lists for a complete document. It is a ClassifyIt profile - the official Office profile of ADatP-4778.2 is XML only. Only finished, closed files are signed: by Explorer's Sign Classification and Classify File, and by Outlook for Office attachments when an email is sent (the attached copy only; never blocks the send). Office writes the parts only when it saves, so saving in Office stores the bare label again. Requires "stanag4774" and the user's own certificate.

With "stanag4774Policy" NATO the label follows the NATO Security Policy (ADatP-4774, Appendix 2): it carries the mandatory Context category with the value of "stanag4774Context" (and the value Releasable when the document is released), the release category is named Releasable To, the categories are named Additional Sensitivity and Administrative, and every category carries its URI. Without "stanag4774Context" no label is written; the Classify It log records it.

The binding's hash covers the original file, before compression and encryption; the binding names the CI-TDF specification version as its profile. Containers without a signing certificate get a binding that is not cryptographically protected.

The configuration owner is responsible for the policy, classification and category names matching the governing security policy; ClassifyIt writes them as configured. If other systems are to check the labels, use the exact names of the policy's SPIF in "real" and "beforeReleaseStrings". Sign the configuration again after changing it. The format is described in the CI-TDF specification (CI-TDF_ManifestSpecification.md).

Article XV. iCLASS

iCLASS (International Classification) is the structured classification value used by ClassifyIt for machine-readable handling of classified information, and codes for human recognition. The visible text shown to users can be local, translated or policy-specific, but the iCLASS value should remain stable so that documents, emails, DLP systems and other receiving systems can interpret the marking consistently.

For administrators, the relevant configuration point is the property key, normally "iCLASS", defined by classifyit_control.propertyKeyName. ClassifyIt writes a JSON value containing the originator code ("org" - the organisation that applied the marking), the classification code ("class") and the release codes ("rel") to document properties and email headers. Optional fields (new in 7.1): "cat" the categories and "only", "rev" the review date, "suc" the downgrade (with "date", "class" and "rel") and, in CI-TDF containers, "dat" the classification date. These values are used by downstream controls such as DLP, mail gateways, XML/application guards and other ClassifyIt installations.

For information managers, iCLASS defines the controlled vocabulary behind the user-facing labels. Define the classification codes, classification levels, release codes, optional release groups, translated labels and the default marking before rollout. Classification levels should run from the least protective to the most protective marking, and release codes should identify the permitted recipients or communities. Avoid changing these codes after deployment unless migration and interoperability effects have been assessed.

Before operational deployment, verify that the visible marking, the document property and the email header all contain the expected iCLASS values for representative documents, emails and release-marking combinations.

PUBLIC (P) and UNCLASSIFIED (U) carry no release: a release given with them is ignored, in markings, in the configuration and in CI-TDF containers.

ClassifyIt takes consistent and extensive use of iCLASS, as it is used for technical key-value pairs, representing the classification of a document or email, condensed versions of display for paragraph markings in word documents, and inclusion in TDF encrypted payload containers adding the iCLASS structure to the manifest.