

# CIFOM-Master



# & CIFOM-Spy



**Documentations des logiciels CIFOM-Spy et CIFOM-Master  
Logiciels de surveillances à distance.**

GIRARDET Nicolas, 4CIN2, Salle 226, CIFOM-ET, Le Locle, avril – mai 2006

## Licence

Copyright (c) 2006 Girardet Nicolas (nicolas.girardet@gmail.com).  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## GNU Free Documentation License

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any

mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only. The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to

be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## **5. COMBINING DOCUMENTS**

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## **6. COLLECTIONS OF DOCUMENTS**

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from

you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## **10. FUTURE REVISIONS OF THIS LICENSE**

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## Tables des matières

I. <b>Licence</b> .....	<b>2</b>
II. <b>GNU Free Documentation License</b> .....	<b>2</b>
0. PREAMBLE.....	2
1. APPLICABILITY AND DEFINITIONS .....	2
2. VERBATIM COPYING.....	4
3. COPYING IN QUANTITY .....	4
4. MODIFICATIONS .....	5
5. COMBINING DOCUMENTS .....	6
6. COLLECTIONS OF DOCUMENTS.....	7
7. AGGREGATION WITH INDEPENDENT WORKS .....	7
8. TRANSLATION.....	7
9. TERMINATION.....	7
10. FUTURE REVISIONS OF THIS LICENSE.....	8
III. <b>Tables des matières</b> .....	<b>9</b>
IV. <b>Introduction</b> .....	<b>12</b>
V. <b>Documentation développeur</b> .....	<b>13</b>
La base de données.....	13
Modèle conceptuel de données.....	13
Modèle logique de données.....	13
Modèle physique de données .....	14
Application WEB CIFOM-Master .....	16
Carte du site .....	16
Schéma synoptique.....	17
Maquettes .....	17
<i>Page de surveillance</i> .....	17
<i>Page des statistiques</i> .....	18
<i>Page de login</i> .....	18
<i>Page de configuration des utilisateurs</i> .....	18
Envois et réceptions de formulaires .....	19
<i>Réalisation d'un formulaire</i> .....	19
Description des pages principales, classes et fonctions utilisées.....	20
<i>La sécurité</i> .....	20
<i>conf.inc.php</i> .....	20
<i>constantes.inc.php</i> .....	20
<i>fonctions.inc.php</i> .....	20
<i>Classe sql_db</i> .....	21
<i>interface.inc.php</i> .....	22
<i>stats.inc.php</i> .....	23
Dialogue entre C# et MySQL.....	25
<i>Schéma de dialogue</i> .....	25
Explication de la page « spy/common.inc.php » .....	25
<i>Commande « refresh »</i> :.....	25
<i>Commande « nomUpdate »</i> .....	25
<i>Commande « msgUpdate »</i> .....	26
<i>Commande « srcUpdate »</i> .....	26

<i>Commande « dateUpdate »</i> .....	26
<i>Commande « appliUpdate »</i> .....	26
Explication de la page spy/update.inc.php .....	27
<i>Contrôle d'une mise à jour</i> .....	27
Application C# CIFOM-Spy .....	28
Carte de l'application .....	28
Schéma synoptique.....	28
Maquettes .....	29
<i>WinForm (Form1.cs)</i> .....	29
<i>spyBlock (SpyBlock.cs)</i> .....	30
<i>setSrv (setSrv.cs)</i> .....	31
Classes, méthodes et propriétés utilisées .....	31
<i>Classe WinForm (Form1.cs)</i> .....	31
<i>Classe SingleInstanceApp (Form1.cs)</i> .....	46
<i>Classe spyBlock (SpyBlock.cs)</i> .....	46
<i>Classe setSrv (setSrv.cs)</i> .....	47
<i>Classe Fonction_Xor (Fonction_Xor.cs)</i> .....	48
<i>Classe FTPFactory (reqFTP.cs)</i> .....	48
<i>Classe ReqHttp (ReqHttp.cs)</i> .....	52
<i>Classe WindowsDesktop (WindowsDesktop.cs)</i> .....	54
Application updateSpy .....	55
Carte de l'application .....	55
Schéma Synoptique .....	55
Maquettes .....	55
<i>frmUpdate (form1.cs)</i> .....	55
Classes, méthodes et propriétés utilisées .....	55
<i>Classe frmUpdate (form1.cs)</i> .....	55
<i>Classe Fonction_Xor (Fonction_Xor.cs)</i> .....	57
<i>Classe ReqHttp (ReqHttp.cs)</i> .....	57
<b>VI. Documentation utilisateur .....</b>	<b>58</b>
Installation du CIFOM-Master .....	58
Le serveur WEB .....	58
<i>Configuration</i> .....	58
<i>Ajout de la base de données</i> .....	60
<i>Copie de l'application WEB</i> .....	61
Le serveur FTP .....	61
<i>Configuration</i> .....	61
Utilisation du CIFOM-Master .....	64
Page de connexion .....	64
Page de configuration .....	64
Page de surveillance .....	64
<i>Signification des couleurs</i> .....	65
<i>Actualiser</i> .....	66
<i>Blocage/Déblocage</i> .....	66
<i>Détails tâches</i> .....	67
<i>Copie d'écran</i> .....	67
<i>Stop tâches</i> .....	67
<i>Message ciblé</i> .....	67

	<i>Message collectif</i> .....	67
	<i>Réinitialiser</i> .....	68
	<i>Fraude</i> .....	68
	Page des statistiques .....	68
	Mise à jour du CIFOM-Spy .....	69
	Installation du CIFOM-Spy .....	69
	Configuration .....	70
	Mise à jour .....	70
	Programme minimisé .....	70
	Programme maximisé .....	71
VII.	<b>Conclusion</b> .....	<b>72</b>
VIII.	<b>Bibliographie</b> .....	<b>73</b>
IX.	<b>Annexe</b> .....	<b>73</b>

## Introduction

Dans le cadre de ma fin de formation professionnelle à l'école technique du centre intercommunal de formation des montagnes neuchâteloises (CIFOM-ET), il m'a été demandé de réaliser une application Windows® programmée en C# qui récupère et vérifie si les programmes qui sont actuellement exécutés sur la machine cliente sont autorisés ou non et tous ces programmes qui n'existent pas encore dans la base de données MySQL y seront ajoutés. Ainsi qu'une application WEB programmée en PHP.

Le but de ce projet est de pouvoir surveiller plusieurs ordinateurs d'une même salle de cours. Un des points forts est de permettre la configuration dans la base de données MySQL des applications ainsi que les titres d'applications qui sont autorisés ou interdits, afin que le programme C# (nommé CIFOM-Spy) sache lesquelles sont autorisées à être exécutées (les applications interdites seront automatiquement arrêtées).

L'application WEB (nommé CIFOM-Master) permet la surveillance des postes qui exécutent CIFOM-Spy. Grâce à la page WEB de surveillance du CIFOM-Master, on a la possibilité d'interagir avec les machines clientes, voici la liste des actions possibles :

- Voir les applications qui sont exécutées par les ordinateurs.
- Envoyer un message à un ou tous les ordinateurs.
- Bloquer/Débloquer un ou tous les ordinateurs.
- Obtenir le détail d'une application.
- Obtenir une copie d'écran.
- Arrêter une application.

Une page de statistique sélectionnée par date et par N° de poste est également disponible. Celle-ci très complète, affiche le détail trié par ordre chronologique des applications avec leurs titres qui ont été exécutés par un poste donné à une date donnée.

## Documentation développeur

Grâce à cette documentation, le lecteur ayant de bonnes connaissances en PHP, C# et SQL sera à même de pouvoir maintenir le projet et de le faire évoluer si nécessaire.

### La base de données

La base de données du projet doit être stable et correctement conçue car elle est très fortement mise à contribution, elle est destinée à évoluer très vite car des milliers de titres vont transiter entre les postes clients et le serveur MySQL.

#### Modèle conceptuel de données

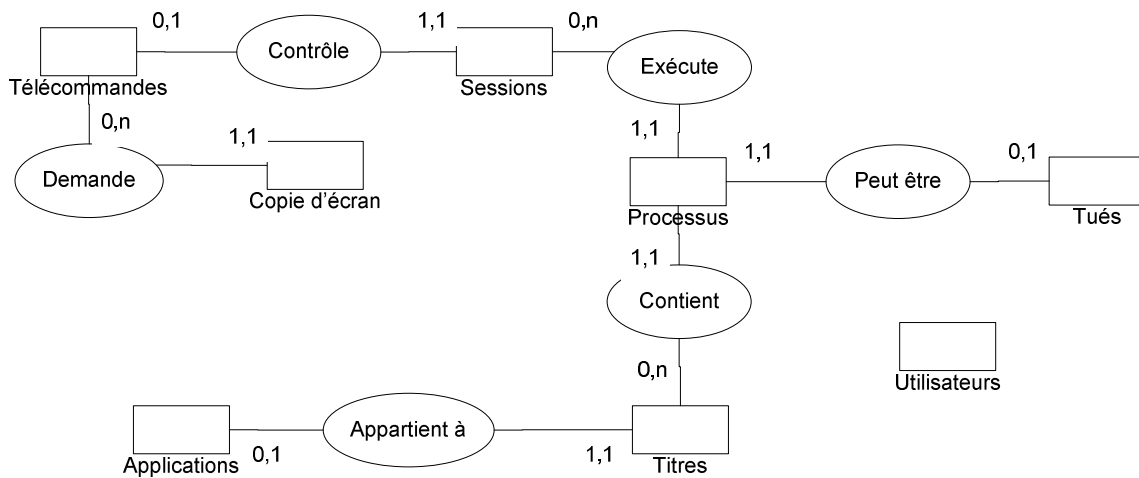


Figure 1 : Modèle conceptuel de données de la base cifomspy

#### Modèle logique de données

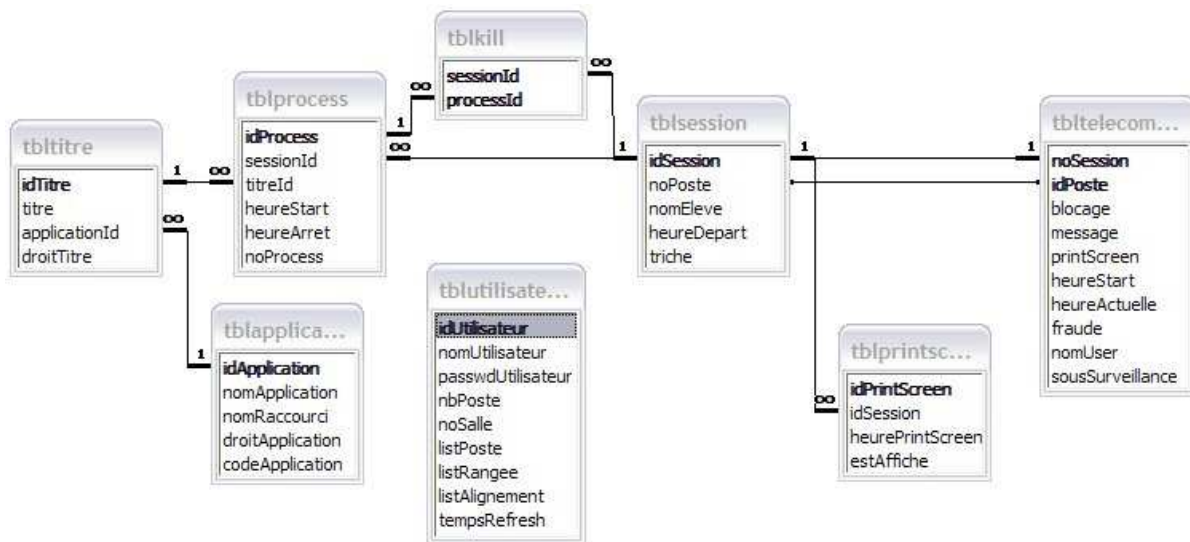


Figure 2 : Modèle logique de données de la base cifomspy

Légende :

- **Gras** : signifie clé primaire.

## Modèle physique de données

### tblapplication

Champ	Type	Null	Défaut
<u>idApplication</u>	mediumint(8)	Non	
nomApplication	varchar(30)	Non	
nomRaccourci	varchar(12)	Oui	NULL
droitApplication	enum('1', '0')	Non	1
codeApplication	int(11)	Non	0

### tblkill

Champ	Type	Null	Défaut
<u>sessionId</u>	mediumint(8)	Non	0
<u>processId</u>	mediumint(8)	Non	0

### tblprintscreen

Champ	Type	Null	Défaut
<u>idPrintScreen</u>	mediumint(8)	Non	
idSession	int(11)	Non	0
heurePrintScreen	datetime	Non	0000-00-00 00:00:00
estAffiche	enum('1', '0')	Non	0

### tblprocess

Champ	Type	Null	Défaut
<u>idProcess</u>	mediumint(8)	Non	
sessionId	mediumint(8)	Non	0
titreId	mediumint(8)	Non	0
heureStart	datetime	Non	0000-00-00 00:00:00
heureArret	datetime	Oui	NULL
noProcess	int(11)	Non	0

### tblsession

Champ	Type	Null	Défaut
<u>idSession</u>	mediumint(8)	Non	
noPoste	int(11)	Non	0
nomEleve	varchar(25)	Non	
heureDepart	datetime	Non	0000-00-00 00:00:00
triche	enum('1', '0')	Non	0

### tbltelecommand

Champ	Type	Null	Défaut
<u>idTelecommand</u>	mediumint(8)	Non	
noSession	int(11)	Non	0
idPoste	mediumint(8)	Non	0
blocage	enum('1', '0')	Non	0
message	varchar(160)	Oui	NULL
printScreen	tinyint(1)	Non	0
heureStart	datetime	Oui	NULL
heureActuelle	datetime	Oui	NULL
fraude	enum('1', '0')	Non	0
nomUser	varchar(30)	Oui	NULL
sousSurveillance	enum('1', '0')	Non	0

## La base de données

### tbltitre

Champ	Type	Null	Défaut
idTitre	mediumint(8)	Non	
titre	varchar(120)	Non	
applicationId	int(11)	Non	0
droitTitre	enum('1', '0')	Non	1

### tblutilisateurs

Champ	Type	Null	Défaut
idUtilisateur	mediumint(8)	Non	
nomUtilisateur	varchar(50)	Non	
passwdUtilisateur	varchar(32)	Non	
nbPoste	mediumint(3)	Non	0
noSalle	varchar(20)	Non	
listPoste	text	Non	
listRangee	text	Non	
listAlignement	text	Non	
tempsRefresh	mediumint(2)	Non	0

Figure 3 : Modèle physique de données de la base cifomspy

## Application WEB CIFOM-Master

CIFOM-Master regroupe deux pages principales ; une page contrôle des postes clients et une page de statistiques. Ces deux pages sont détaillées ci-dessous. La carte du site, le schéma synoptique, les maquettes des pages ainsi que des informations techniques sont également disponible.

### Carte du site

Site - cifomspy (C:\Program Files\EasyPHP1-	Dossier racine.
config	Dossier d'inclusion de la page config.php
config.inc.php	- Réception du formulaire de config.php
files	Dossier des fichiers
cspy_1.5.2339.21858.exe	- Dernière version de CIFOM-Spy
gestion	Dossier des fichiers de commandes.
actualise.inc.php	- Actualisation des champs de contrôles.
block.inc.php	- Commande dé/blocage.
copie.inc.php	- Commande copie d'écran.
details.inc.php	- Commande détail d'une tâche.
init.inc.php	- Réinitialisation des sessions de postes.
msg.inc.php	- Commande envoyer un message.
stop.inc.php	- Commande arrêt d'une tâche.
img	Dossier des copies d'écrans reçu.
PrintScreen_1.bmp	- Copie d'écran ID N°1.
PrintScreen_2.bmp	- Copie d'écran ID N°2.
includes	Dossier des fichiers d'inclusion.
common.inc.php	- Fichier « kernel » du site.
conf.inc.php	- Fichier de configuration du site.
constantes.inc.php	- Fichier contenant le nom des tables.
db.inc.php	- Classe sql.
fonctions.inc.php	- Fonctions diverses du site.
interface	Dossier des fichiers de l'interface du site.
interface.inc.php	- Fichier générant l'interface du site.
login	Dossier des fichiers du login du site.
login.inc.php	- Fichier gérant l'accès utilisateurs.
medias	Dossier des images et CSS du site.
4_04_009_Charte 1024x768.jpg	- Image du fond du site.
styles.css	- Styles CSS du site.
spy	Dossier de liaison CIFOM-Spy ↔ Master
common.inc.php	- Fichier exécutant les requêtes du Spy.
update.inc.php	- Fichier vérifiant les mises à jour du client.
stats	Dossier de la page statistique.
stats.inc.php	- Fichier générant le contenu de la page.
affScreen.php	- Page affichant les copies d'écrans.
config.php	- Page de configuration des utilisateurs.
index.php	- Page d'accueil du site.
login.php	- Page de login du site.
stats.php	- Page des statistiques du site.

Figure 4 : Carte du site de l'application WEB CIFOM-Master

### Schéma synoptique

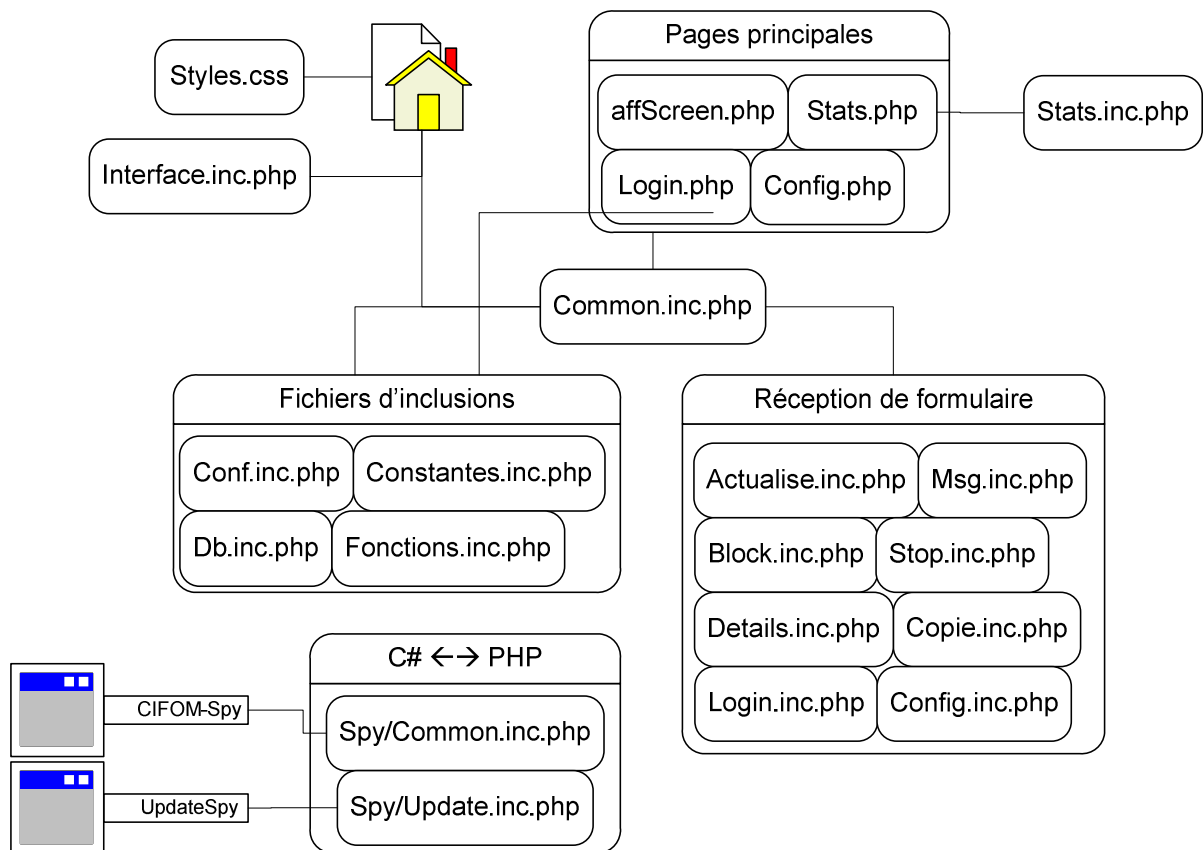


Figure 5 : Schéma synoptique de l'application CIFOM-Master

### Maquettes

#### Page de surveillance

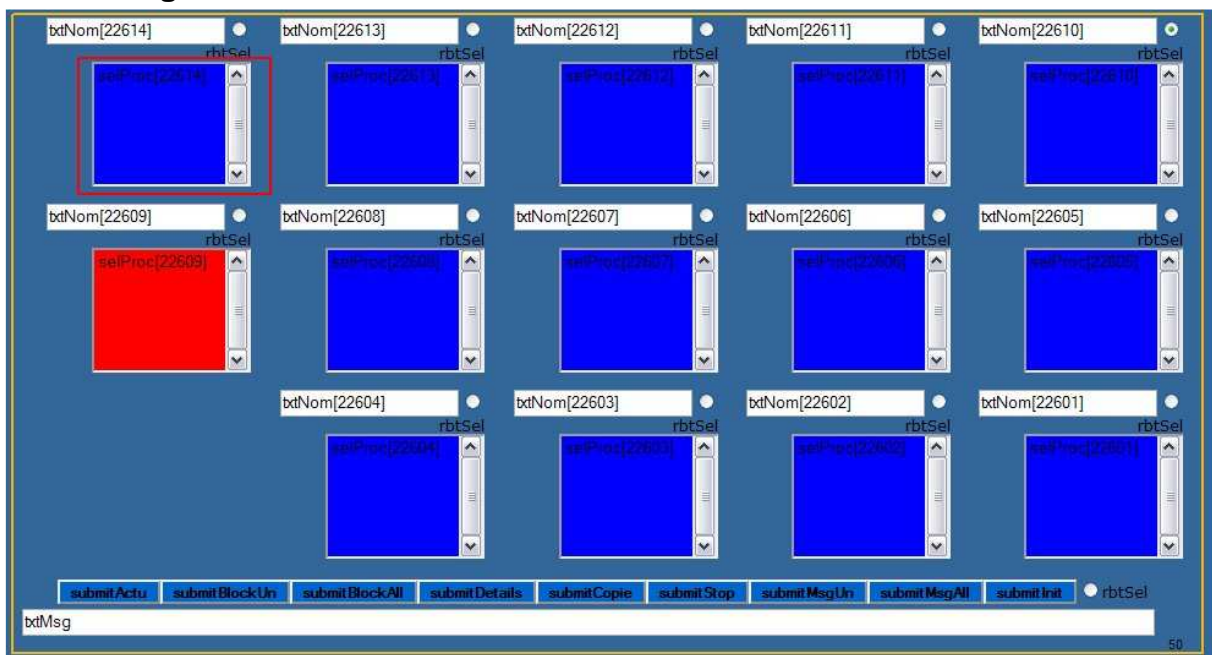


Figure 6 : Maquette de la page de surveillance

#### Explications :

- Le cadre **jaune** représente le contenu généré par la page « *interface.inc.php* ».

- Le cadre rouge représente la liste de sélection générée par la fonction « *GetListProcessByHost* » de la page « *fonctions.inc.php* ».

### Page des statistiques

Poste :  Date :

**10:53 - Mister X - Triché**

**notepad**

11:31	11:35	Sans titre - Bloc-notes
11:31	11:45	é-ö-Ã -ä-S-£-è-&utilde;-î-â-ê-û-.txt - Bloc-notes

**iexplore**

11:31		CIFOM - Centre interrégional de formation des Montagnes neuchâtelaises - Microsoft Internet Explorer
-------	--	--

**ssmarque.scr**

11:40	11:40	Screen Saver
-------	-------	--------------

Figure 7 : Maquette de la page statistique

### Explications :

- Le cadre gris représente le contenu généré par la page « *stats.inc.php* ».

### Page de login

**Formulaire de connexion au panneau d'administration**

Nom d'utilisateur :

Mot de passe :

Aller à :

Figure 8 : Formulaire de connexion.

### Page de configuration des utilisateurs

**Formulaire de configuration**

**Aller à : [Page de surveillance](#) - [Page de statistiques](#)**

Nom d'utilisateur\* :

Mot de passe :  (Laisser vide si inchangé).

N° de salle\* :

Nombre de poste\* :

Liste des postes de la salle\* :   
(Séparer par des ';' Ex.: 01;02;03)

Nombre de poste par rangées\* :   
(Séparer par des ';' Ex.: 1;2)

Alignement des postes par rangées\* :   
(Séparer par des ';' Ex.: 'r','l') l = gauche, r = droite

Temps de rafraîchissement\* :  (Rafraîchissement en secondes de la page surveillance)

\* Paramètres obligatoires.

### Principe de fonctionnement :

Si nous avons trois postes dans la salle N°226, ces deux ordinateurs sont organisés en deux rangées, la première rangée étant celle contenant le plus d'ordinateur, son alignement importe peu. Par contre la seconde rangée ayant un poste de moins, nous pouvons décider d'aligner ce poste sur la droite de la page. Voici la configuration à réaliser :

#### Configuration

```
nosalle = 226
nbposte = 3
listeposte = 01;02;03
listerangee = 2;1
alignrangee = 1;r
```

#### Résultat :



Figure 9 : Résultats selon la configuration du principe de fonctionnement.

### Envois et réceptions de formulaires

Afin de mieux gérer et sécuriser les données envoyées par formulaire, une stratégie a été mise en place. Pour cet exemple, le formulaire de la page login va être reproduit.

#### Réalisation d'un formulaire

Pour garder la hiérarchisation mise en place, lorsque vous réalisez une nouvelle page (avec ou sans formulaire), placez celui-ci à la racine du site.

Balise « *form* » de la page « *login.php* »

```
<form name="frmLogin" method="post" action="<? echo $_SERVER['PHP_SELF']; ?>">
//Contenu du formulaire
    <input type="submit" name="submitLogin" value="Connexion" />
</form>
```

Le nom du bouton de type « *submit* » est très important car celui-ci permettra à la page « *common.inc.php* » de savoir quel formulaire a été envoyé pour charger la page de réception qui convient comme expliqué ci-dessous.

Modification de la page « *common.inc.php* »

```
elseif(!empty($_POST['submitLogin']))
    include("../config/login.inc.php");
```

En incluant la page « *login.inc.php* » la première chose à faire c'est le traitement des caractères.

Traitement des caractères de la page « *login.inc.php* »

```
/*Partie sécurisant les données reçues par formulaire*/
foreach($_POST as $key => $val) //Parcours toutes les valeurs
    $_POST[$key] = htmlentities(trim($val)); //Remplace les caractères spéciaux.
extract($_POST); //Extrait les clés pour en créer des variables.
```

Après ces quelques étapes, vous pouvez effectuer vos instructions.

## Description des pages principales, classes et fonctions utilisées.

Afin de mieux comprendre la génération de la page d'index ainsi que celle de la page des statistiques, des informations techniques décrivent les parties les plus complexes.

### La sécurité

Toutes les pages principales tels que « *index.php* » et « *stats.php* » doivent contenir en haut du code source les lignes suivantes :

```
<?
    if(empty($_COOKIE['spyMaster']))
        header("location: login.php");
    //Définis une constante requise pour charger les pages d'inclusions.
    define("IN_WEBSITE", TRUE, TRUE);
    include("./includes/common.inc.php"); //Page traitant les inclusions.
?>
```

Le code ci-dessus permet de vérifier si l'utilisateur s'est connecté, si ce n'est pas le cas il est redirigé vers la page login.

Si il est logué, la constante « *IN\_WEBSITE* » est défini avec la valeur « *TRUE* », cette constante est utilisée par toutes les pages de type « *\*.inc.php* » (voir ci-dessous) afin d'empêcher que celles-ci ne soient chargées sans autorisation.

```
//Si la page appelante n'a pas déclaré la constante IN_WEBSITE.
if(!defined("IN_WEBSITE"))
    die(""); //Interruption du chargement de la page.
```

### conf.inc.php

Ce fichier vous permet de configurer les paramètres d'accès à la base de données :

\$serveur	Nom du serveur MySQL.
\$nomuser	Nom d'utilisateur de la base MySQL.
\$motpass	Mot de passe de la base MySQL.
\$nombase	Nom de la base de données.
\$serveurName	Adresse du serveur WEB.

### constantes.inc.php

Ce fichier contient toutes les variables constantes du site, tel que le nom des tables de la base de données afin de faciliter la modification du site si on devait changer le nom de l'une d'entre elles.

### fonctions.inc.php

Les quatre fonctions du site créées par moi-même sont disponible dans ce fichier voici une brève description de chacune d'elles :

- **GetListProcessByHost**
  - But : Retrouvé la liste des processus pour un poste donné.
  - Entrée : l'ID du poste
  - Sortie : La liste des processus sous forme de liste de choix (HTML).
- **GetSessionsKeys**
  - But : Retrouvé la clé d'un poste donné figurant dans le tableau `Pc_Control`.
  - Entrée : l'ID du poste.
  - Sortie : La clé.

- **ArrayFilter**
  - But : Retrouvé la plus grande valeur d'un tableau donné.
  - Entrée : Le tableau.
  - Sortie : La plus grande valeur.
- **SpecialCarac**
  - But : Remplacer certains caractères spéciaux tel que : é,à,è,...
  - Entrée : Le texte à modifier.
  - Sortie : Le texte modifié.
- **StripCaracteres**
  - But : Remplacer les caractères ` ;` et ` !` par `\\\_` et `//\_`.
  - Entrée : Le texte à modifier.
  - Sortie : Le texte modifié.
- **UnstripCaracteres**
  - But : Remplacer les caractères `\\\_` et `//\_` par ` ;` et ` !`.
  - Entrée : Le texte à modifier.
  - Sortie : Le texte modifié.

### **Classe sql\_db**

Dans le fichier « db.inc.php », vous trouverez la classe « sql\_db ». C'est une classe qui permet de reproduire les fonctions « mysql » standard.

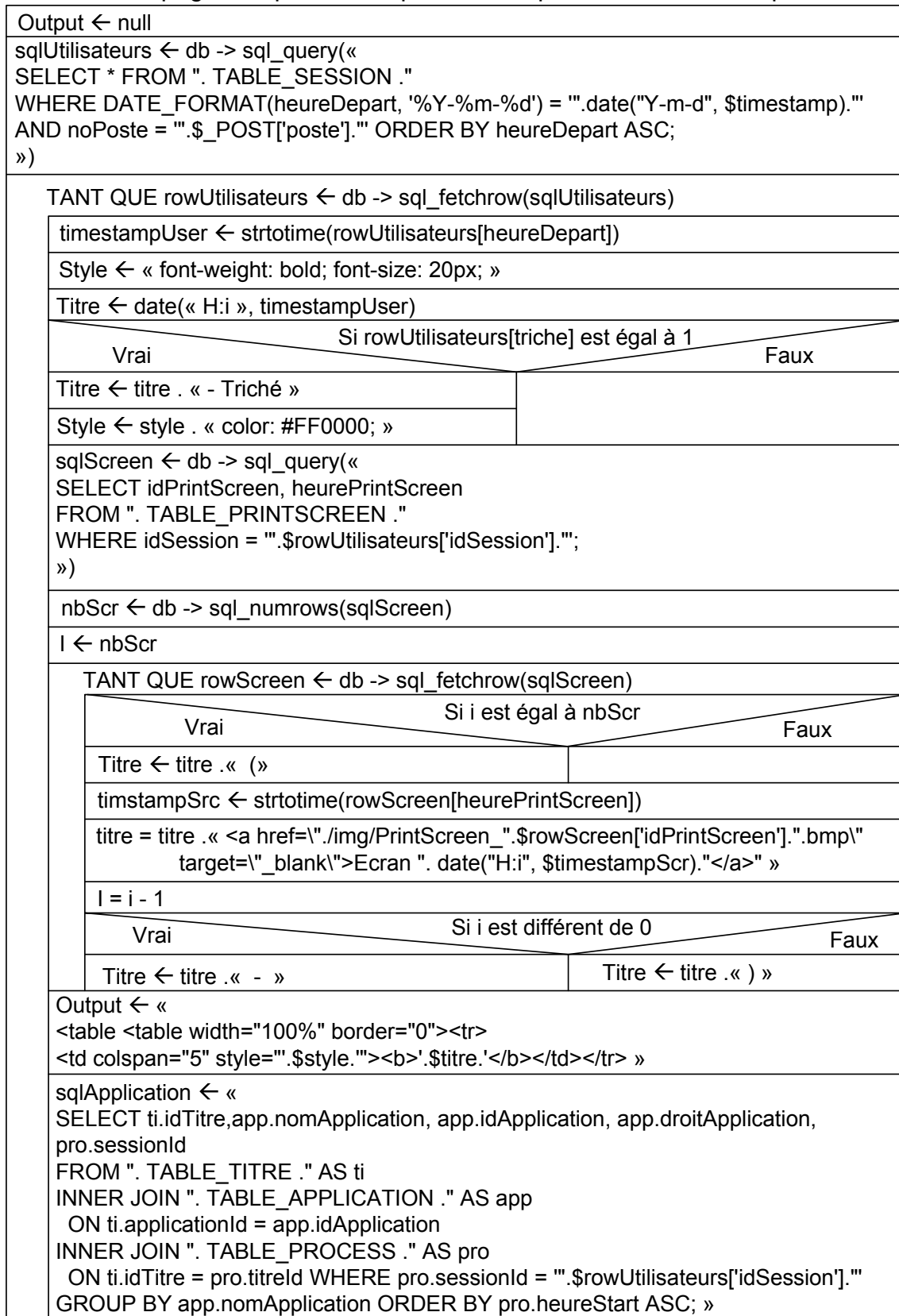
Classe : sql\_db

sql_db.....	Constructeur
sql_close.....	Libère le buffer et ferme la connexion à la base.
sql_query.....	Exécute une requête.
sql_numrows.....	Retourne le nombre d'enregistrement affecté par la dernière requête de sélection.
sql_affectedrows.....	Retourne le nombre d'enregistrement affecté par la dernière requête.
sql_numfields.....	Retourne le nombre de champs concerné par la dernière requête de sélection.
sql_fieldname.....	Retourne le nom d'un champ donné de la dernière requête de sélection.
sql_fieldtype.....	Retourne le type d'un champ donné de la dernière requête de sélection.
sql_fetchrow.....	Retourne les enregistrements de la dernière requête de sélection (1 à 1).
sql_fetchrowset.....	Retourne les enregistrements de la dernière requête de sélection (tous).
sql_fetchfield.....	Retourne les champs concerné par la dernière requête de sélection.
sql_fetchxml.....	Retourne le résultat de la dernière requête de sélection au format XML.
sql_rowseek.....	Déplace le pointeur de la dernière requête de sélection.
sql_nextid.....	Retourne l'ID de la dernière requête d'insertion.



**stats.inc.php**

Cette page est appelée par « stats.php » au moment de la génération du contenu centrale de la page lorsqu'un N° de poste ainsi qu'une date ont été spécifiés.



**Figure 11 : Structogramme de la page "stats.inc.php" (1<sup>ère</sup> partie)**

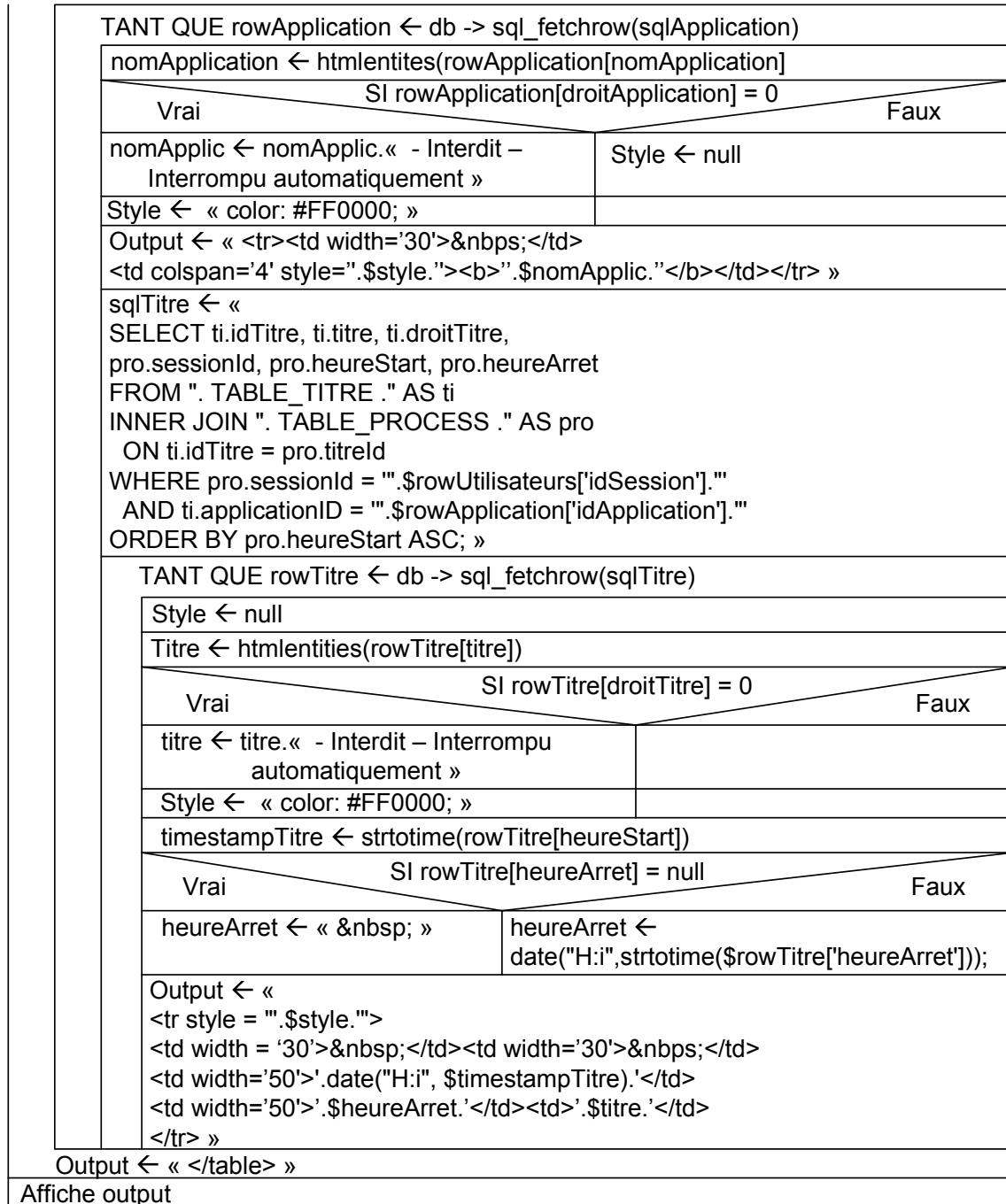


Figure 12 : Structogramme de la page "stats.inc.php" (2ème partie)

## Dialogue entre C# et MySQL

Pour la communication entre le CIFOM-Spy et la base de données MySQL, une page PHP (« *common.inc.php* ») a été réalisée pour exécuter les requêtes du CIFOM-Spy. Nous passons par une page PHP car la sécurité de MySQL empêche un utilisateur de s'y connecter si celui-ci ne dispose pas de l'IP 127.0.0.1 ou si son adresse IP ne figure pas dans la liste des adresses autorisées.

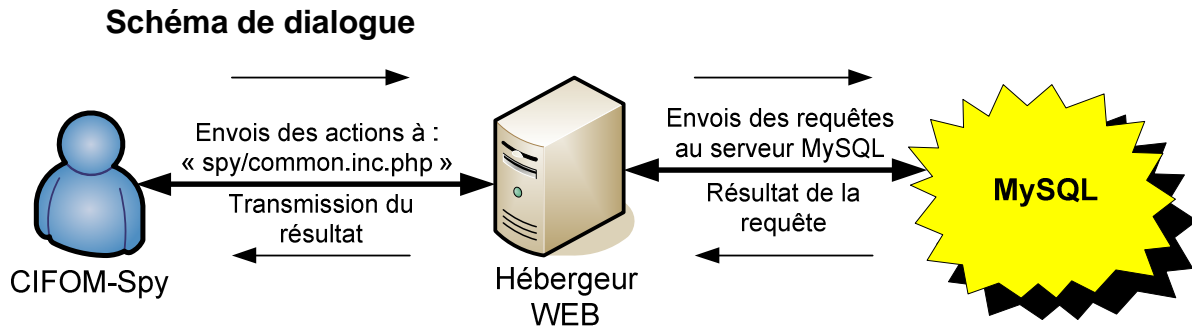


Figure 13 : Schéma de communication C# et MySQL

## Explication de la page « *spy/common.inc.php* »

Lorsque le client envoie des données, en méthodes POST voici un détail des commandes ci-dessous :

### Commande « refresh » :

Lorsque le client envoie cette commande, c'est qu'il désire mettre à jour ses données. Il aimerait savoir si il a des actions à effectuées tel qu'afficher un message, bloquer l'écran, ...

### Paramètres reçus :

Nom	Valeur
timedepart	Heure de départ du client.
nom	Nom du poste

### Paramètres envoyés

Nom	Valeur
noSession	Le N° de session récupéré depuis la base de données.
blocage	1 si on doit bloquer le poste, 0 sinon.
message	Le message si il y'en a un à transmettre.
printScreen	L'ID de la copie d'écran si il y'en a un à faire.
heureStart	L'heure de départ du client.
nomUser	Le nom d'utilisateur de la session.
sousSurveillance	1 si le poste est sous surveillance, sinon 0.

### Commande « nomUpdate »

Elle est effectuée lorsque le client constate que le nom qu'il a enregistré et le nom de la base de donnée à changé.

### Paramètres reçus :

Nom	Valeur
nomUser	Le nom d'utilisateur de la session.
nosession	Le N° de session.

**Commande « msgUpdate »**

Met le champ Message à NULL dans la base de données, cela signifie également que le client à afficher le message.

**Paramètres reçus :**

Nom	Valeur
nosession	Le N° de session.

**Commande « srcUpdate »**

Lorsque le client a envoyé sa dernière copie d'écran sur le serveur FTP, il envoi l'ID pour indique au Master que l'image a été envoyée.

**Paramètres reçus :**

Nom	Valeur
prtScr	L'ID de la copie d'écran.
nosession	Le N° de session.

**Commande « dateUpdate »**

Met à jour la date et l'heure de la dernière vérification des données du CIFOM-Spy.

**Paramètres reçus :**

Nom	Valeur
date	La date et l'heure actuelle
nosession	Le N° de session.

**Commande « appliUpdate »**

Mise à jour diverses concernant les applications.

**Paramètres reçus :**

Action	Nom	Valeur
recup	-	-
add	name	Le nom de l'application
check	titres	Les titres à faire vérifier.
kill	appld	L'ID de l'application tuée.
udp	process	La liste des processus de la machine.
udpKill	nosession	Le N° de session.
stopPro	process	Les IDs des titres.
	nosession	Le N° de session.

**Paramètres envoyés :**

Action	Valeur
recup	La liste des applications autorisées et interdites.
add	L'Id de l'application qui vient d'être ajoutée.
check	La liste des IDs des titres vérifiés..

kill	-
udp	-
udpKill	Les N°des processus systèmes à quitter.
stopPro	-

### Explication de la page spy/update.inc.php

Avant que le client ne fasse sa demande d'ouverture de session, il exécute le processus « updateSpy.exe ».

UpdateSpy va envoyer une requête à la page spy/update.inc.php en lui transmettant la version actuelle de cifomSpy.exe.

### Contrôle d'une mise à jour

En recevant la version de cifomSpy.exe, la page va scanner le dossier « files » en excluant les fichiers : « thumbs.db », « updateSpy.exe », « cifomSpy.exe ». A partir des fichiers que la page « update.inc.php » va trouver, elle extraira le n°de version des fichiers. Pour ce faire, il faut que le nom de fichier soit sous la forme :

« cspy\_w.x.y.z.exe », les valeurs w,x,y,z sont additionnées et comparées à l'addition des valeurs de la version envoyées par « updateSpy.exe ».

### Exemple :

Si la version envoyée par « updateSpy.exe » est 1.5.2345.23344 l'addition de ces valeurs sera : 25695.

Si le fichier dans le dossier « files » se nomme « cspy\_1.5.2345.23345.exe » l'addition de ces valeurs sera : 25696.

Le fichier du dossier « files » disposant d'un plus grand résultat cela signifie que le client ayant fait la demande n'est pas à jour.

La page « update.inc.php » va donc retourner « 1 !<Nom complet du fichier> ! <Version du fichier> ! <Taille du fichier> ». Dans le cas où le client est à jour, la page « update.inc.php » va retourner « 0 ».

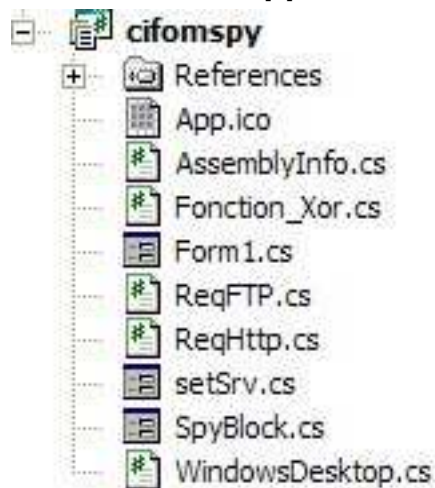
## Application C# CIFOM-Spy

Le but de cette application C# est de pouvoir faire l'interaction entre le CIFOM-Spy et le CIFOM-Master.

Le transfert des données se fait par méthode « POST », jusqu'au serveur WEB qui héberge le CIFOM-Master, à la page « spy/common.inc.php ».

Une des plus grosses difficultés dans la réalisation du programme était la gestion des applications et des titres autorisés et interdits.

### Carte de l'application



Dossier racine.

Dossier des références de l'application..

- Icône de l'application.

- Information d'assemblage.

- Classe « FonctionXOR ».

- Formulaire principale.

- Classe « ReqFTP ».

- Classe « ReqHTTP ».

- Formulaire de modification des clés registre.

- Formulaire de blocage de l'écran.

- Classe WindowsDesktop.

### Schéma synoptique

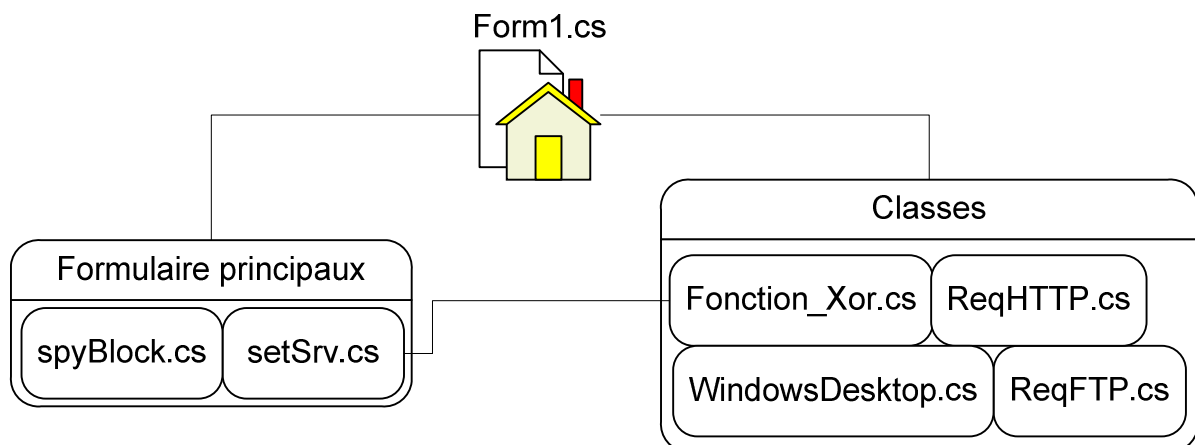


Figure 14 : Schéma synoptique de l'application.

## Maquettes

### WinForm (Form1.cs)



Figure 15 : Maquette du formulaire principale "WinForm".



Figure 16 : Composants cachés de WinForm.

#### Détails :

Objet	Propriété	Valeur
IblNom	Text	Nom
edtNom	Text	NULL
	ReadOnly	True
IblMachine	Text	Machine
edtMachine	Text	NULL
	ReadOnly	True
IblID	Text	ID
edtID	Text	NULL
	ReadOnly	True
IblHeure	Text	Démarré il y a
edtTime	Text	NULL
	ReadOnly	True
IblMinutes	Text	Minutes
btKill	Text	Stopper
	BackColor	Red
	Font	Microsoft Sans Serif; 8.25pt; style=Bold
IblTitre	Text	Titre
edtTitre	Text	NULL
	ReadOnly	True

WinForm	Text	CIFOM Class Controller
	Icon	CIFOMSpy.ico
	MinimizeBox	False
	MaximizeBox	False
	FormBorderStyle	FixedSingle
timer_Refresh	Interval	432000
	Enabled	False
timer1	Interval	2000
	Enabled	True
notifyIcon	Text	CIFOM Spy – Interdit d'arrêter ce programme (AJE)
	Icon	CIFOMSpy.ico
	ContextMenu	contextMenu1
	Visible	True

**spyBlock (SpyBlock.cs)**



Figure 17 : Maquette du formulaire spyBlock

**Détails :**

Objet	Propriété	Valeur
IblBlock	Text	Le professeur vous a bloqué.  Ecoutez attentivement ses explications en attendant qu'il vous redonne le contrôle.
	TextAlign	MiddleCenter
	Font	Microsoft Sans Serif; 12pt; style=Bold
	Dock	Fill
spyBlock	Text	Ecran de blocage
	FormBorderStyle	None

	WindowState	Maximized
	ControlBox	False
	MaximiseBox	False
	MinimizeBox	False

**setSrv (setSrv.cs)**

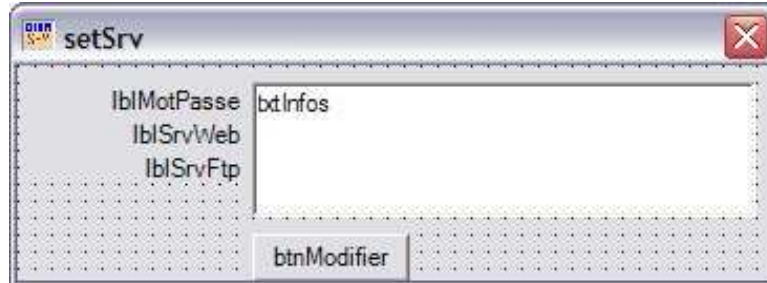


Figure 18 : Maquette du formulaire setSrv

**Détails :**

Objet	Propriété	Valeur
setSrv	Text	Edition des clés de registre – CIFOM-Spy
	MaximumSize	368 ;136
	MinimumSize	368 ;136
	Size	368 ;136
	Icon	CIFOMSpy.ico
	MaximizeBox	False
	MinimizeBox	False
	StartPosition	CenterScreen
IblMotPasse	Text	Mot de passe :
IblSrvWeb	Text	Serveur WEB :
IblSrvFtp	Text	Serveur FTP :
txtInfos	Text	NULL
btnModifier	Text	Modifier

**Classes, méthodes et propriétés utilisées**

**Classe WinForm (Form1.cs)**

**Liste des méthodes et propriétés utilisées :**

Classe	Méthodes/Propriétés	Description
Application	Exit	Quitter l'application
	Run	Démarre l'application
ASCIIEncoding	GetString	Convertit une chaîne de type « <i>byte[]</i> » en « <i>string</i> »
DateTime	Now	Date et Heure de l'instant présent
Directory	CreateDirectory	Créer un dossier.
	Exists	Contrôle si un répertoire existe ou non.
Environment	DoEvents	Exécute toutes les actions en attente, dans la queue.

Application WEB CIFOM-Spy

	MachineName	Nom NetBIOS de la machine.
	StartupPath	Chemin d'exécution de l'application.
FTPFactory	close	Se déconnecte du serveur.
	setDebug	Indique le mode de travail
	setRemoteHost	Adresse du serveur FTP distant.
	setRemoteUser	Nom d'utilisateur autorisé
	setRemotePass	Mot de passe de l'utilisateur.
	setRemotePort	Port sur lequel se connecter.
	upload	Envoi un fichier
Hashtable	Add	Ajoute un enregistrement.
	Clear	Efface tous les enregistrements.
	ContainsKey	Contrôle s'il contient la clé indiquée.
	ContainsValue	Contrôle s'il contient la valeur indiquée.
	Count	Compte le nombre d'enregistrement qu'il contient.
	GetEnumerator	Permet de parcourir la « <i>Hashtable</i> » et extrait les clés et les valeurs.
	Remove	Supprime un enregistrement.
IDictionaryEnumerator	Key	Retourne la valeur de la clé.
	MoveNext	Se déplace jusqu'à la prochaine clé.
	Value	Retourne la valeur associée à la clé.
MessageBox	Show	Ouvre une fenêtre de type message d'avertissements, informations, ....
Process (System.Diagnostics)	GetProcess	Récupère la liste des processus actifs.
	GetProcessById	Récupère un process selon son ID.
	GetProcessesByName	Récupère tous les process selon un nom.
	Id	N° d'identification du process.
	Kill	Force le processus à

Application WEB CIFOM-Spy

		s'arrêter.
	Length	Nombre de process récupéré.
	MainWindowTitle	Titre de la fenêtre de l'application.
RegistryKey	ProcessName	Nom de l'application.
	Close	Ecris les modifications sur le disque.
	LocalMachine.OpenSubKey	Retourne une sous-clé (lecture seule).
	GetValue	Retourne la valeur d'une clé.
	SetValue	Défini/Modifie la valeur d'une clé.
ReqHttp	FullUrl	Adresse complète de la page à laquelle envoyer les données.
	HttpArg httpRequest	Arguments à envoyer. Envoi les données.
String	IndexOf	Renvoie la position de la première occurrence d'un/d'une suite de caractères.
	Insert	Insert une chaîne de caractères dans un texte.
	Length	Renvoie le nombre de caractère d'un texte.
	Remove	Supprime une partie du texte.
	Split	Retourne un tableau dont un texte a été séparé en fonction d'un caractère.
	Substring	Renvoie une partie d'un texte.
	Trim	Supprime les espaces avant le premier et après le dernier caractère.
TextBox	Text	Texte de la TextBox
Thread	Start	Démarre le thread
Timer	Interval	Modifie le temps (en milliseconde) entre chaque « tick » de l'objet.
	Start	Démarre le timer.
	Stop	Arrête le timer
TimeSpan	Minutes	Minutes écoulées
	Hours	Heures écoulées
TreeNode	-	Identique à TreeView
TreeView	BeginUpdate	Stoppe le repaint
	EndUpdate	Reprends le repaint

## Application WEB CIFOM-Spy

	ExpandAll	Développe tous les nœuds.
	Nodes	Collections de nœuds
	Nodes.Add	Ajout d'un nœud
	Nodes.Clear	Suppression de la collection de nœuds
	SelectedNode	Propriété pour sélectionner un nœud.
	SelectedNode.Text	Texte du nœud sélectionné.
WindowsDesktop	CaptureScreen	Fait une copie d'écran.
	save	Enregistre l'image.
WinForm	Capture	Vérifie si la souris se situe sur le formulaire.
	Close	Ferme le formulaire.
	Focus	Donne le focus à un formulaire.
	ShowDialog	Ouvre un formulaire.
	ShowInTaskbar	Affiche la fenêtre dans la barre des tâches.
	WindowState	Définit l'état de la fenêtre (Min/max/tâche).

### Liste des évènements :

Objet	Méthode	Description
btKill	btKill_Click	Clique sur le bouton « Stopper ».
contextMenu1	menultem_About	Choix dans le menu contextuel de la ligne « A propos de ».
	menultem_Maximize	Choix dans le menu contextuel de la ligne « Agrandir ».
Control	WndProc	Redéfinition de l'évènement « System.Shutdown ».
WinForm	WinForm_Load	Chargement du formulaire.
	WinForm_Closing	Fermeture du formulaire.
	WinForm_Resize	Redimensionnement de la fenêtre (minimize/maximize).
timer1	timer1_Tick	Déclenché par le « Tick » du timer1.
timer_Refresh	timer_Refresh_Tick	Déclenché par le « Tick » du timer_Refresh.
treeView1	treeView1_After_Select	Choix d'un nœud dans le TreeView.

## **Explications des évènements :**

### btKill\_Click

Essaye de récupérer l'ID du processus sélectionné. Si ça marche : tue le processus. Dans tous les cas, efface les champs de détail du processus sélectionné.

### menuItem\_About

Affiche un MessageBox avec les informations du logiciel.

### menuItem\_Maximize

Force l'affichage de la fenêtre en mode normal et active l'affichage dans la barre des tâches.

### WndProc

Contrôle si la fermeture du programme est dû à l'arrêt du système d'exploitation ou si l'utilisateur a cliqué sur la « Croix ».

### WinForm\_Load

Charge la liste des processus avec la méthode « LoadCurrentProcess ».

Récupère le nom NetBIOS et le met dans le champ « edtMachine » et dans l'attribut « strNomMachine ».

Réduit la fenêtre dans le système tray.

L'attribut « strHeureDepart » récupère l'heure de départ avec la méthode « GetDateTime ».

Essaye de récupérer la clé de registre « [HKLM\Software\CifomSpy\cSpyWeb] » et « [HKLM\Software\CifomSpy\cSpyFtp] » et met le résultat dans les attributs « strSrvWeb » et « strSrvFtp ». En cas d'échec, ouvre le formulaire « setSrv » puis on essaye à nouveau de charger les clés de registre.

Instancie la classe « ReqHttp » dans l'attribut « request ».

Instancie le « Thread » sur la méthode « noManager ».

Démarrage de « timer1\_Refresh ».

### WinForm\_Closing

Si l'utilisateur provoque la fermeture, on minimise la fenêtre, on empêche la fermeture de la fenêtre et on réinitialise l'attribut « \_closeClick » à False.

Sinon, on ouvre le formulaire, on l'affiche dans la barre des tâches et on marque une pause pour permettre à l'application de vider la queue.

### WinForm\_Resize

Si on minimise la fenêtre, on cache la tâche. Sinon on la montre.

### timer1\_Tick (Avec structogramme à la page suivante)

Lors du déclenchement de l'événement Tick du timer1 (toutes les 2 secondes), le treeview est effacé et remis à jour. Avant de l'effacer, il faut donc sauvegarder la sélection pour pouvoir resélectionner la même tâche après la mise à jour.

Le principe de sauvegarde est simple : si les quatre premiers caractères du nœud sélectionné forment un nombre, ce nombre est l'ID du process sélectionné, sinon la sélection n'est pas sauvegardée.

Ensuite, on bloque la mise à jour du treeView pour éviter de le redessiner pendant qu'on le reconstruit.

Les données du treeView sont alors effacées et une boucle démarre qui parcourt tous les process. Pour chaque process, on regarde si il n'est pas dans la liste des process du départ. S'il n'y est pas, il faut surveiller ce processus. Dans ce cas on regarde si l'application liée au processus est déjà dans le treeView pour savoir si il faut rajouter un nœud parent. Enfin, le nœud enfant est créé avec le numéro de processus et le titre de sa fenêtre. Finalement, on regarde si le numéro de processus est celui qui était sélectionné avant la mise à jour, et si c'est le cas, on le resélectionne.

Après la boucle, les données du treeView sont à jour il reste donc à développer tous ses nœuds et le repeindre.

Après la mise à jour, si aucun nœud n'est sélectionné, effacer les champs de détail du processus sélectionné.

(Description, méthode et structogramme (voir ci-dessous) conçus par le supérieur professionnel Mr. Jeanmaire Alain)

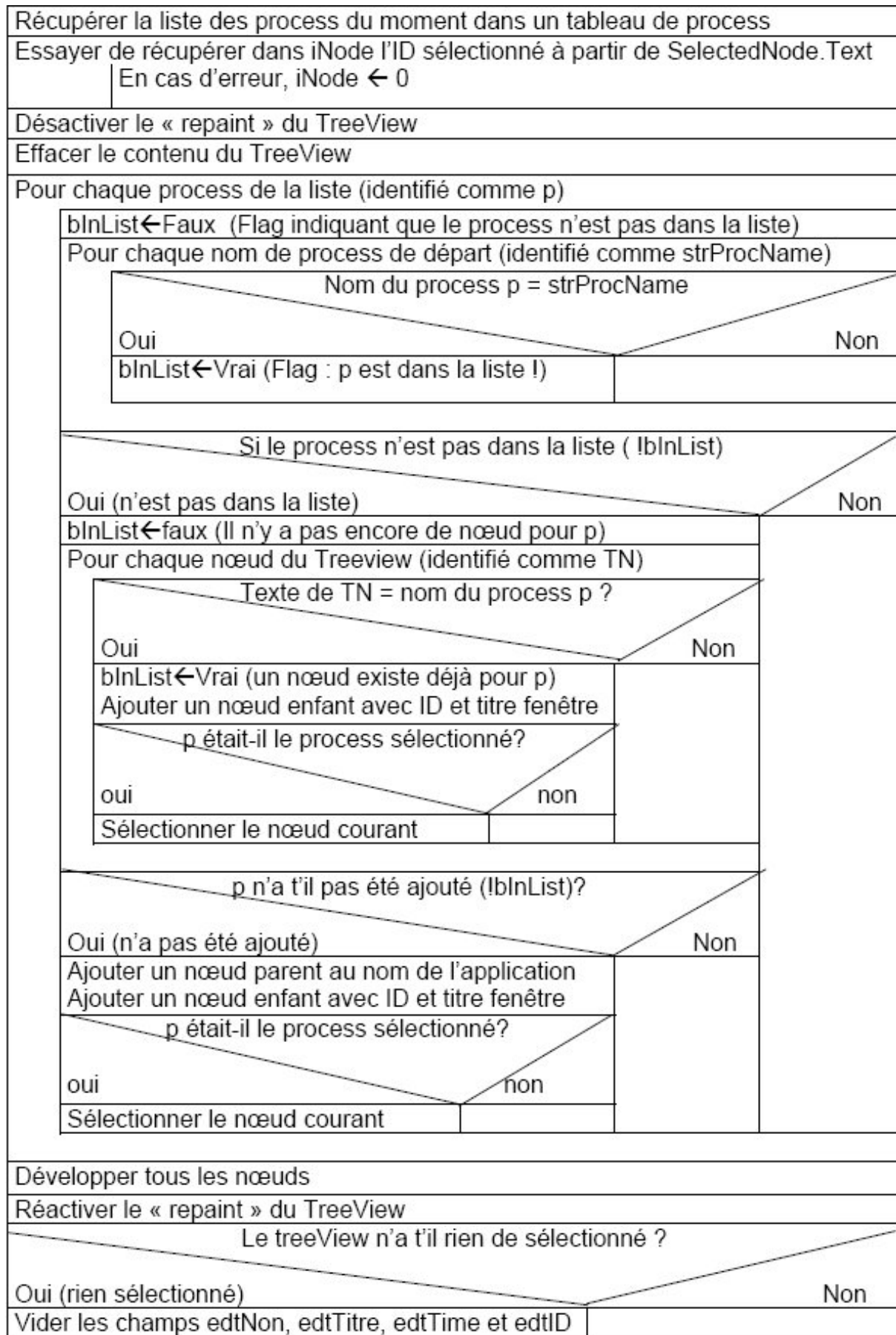


Figure 19 : Structogramme de l'évènement "timer1\_Tick"

timer\_Refresh (Avec structogramme)

Au déclenchement de l'évènement « Tick » de l'objet « timer\_Refresh » (toutes les 2 minutes). On envoie la commande « refresh », l'heure de départ et le nom du poste au serveur WEB à la page « spy/common.inc.php ».

Toute la suite des instructions figure dans un « try – catch » pour éviter des erreurs.

En réponse à notre précédente requête, on récupère le N° de session dans l'attribut « strNoSession », le message (si il y'en a un), l'ID d'une éventuelle copie d'écran, le nom d'utilisateur de la session en cours et si la session est mise sous surveillance par la CIFOM-Master.

Si on est sous surveillance, toutes les instructions suivantes seront exécutées.

Si l'attribut « strNomUser » est NULL et que la variable « dbNomUser » n'est pas NULL, alors on met à jour l'attribut « strNomUser » et on envoie au serveur WEB la commande « nomUpdate » avec en paramètres : le nom d'utilisateur reçu et le N° de session.

Si on doit bloquer l'écran et que le formulaire « SpyBlock » ne capture pas la souris, dans ce cas on contrôle si le système de « Hooking <sup>1</sup> » est installé, si ce n'est pas le cas on démarre le « Thread » thBlock (voir méthode : noManager), on définit l'attribut « blocked » à True, on ouvre le formulaire « SpyBlock » et on modifie l'intervalle de l'objet « timer\_Refresh » à 10 secondes. Dans tous les cas on donne le focus au formulaire « SpyBlock ».

Si on ne doit pas bloquer l'écran, on vérifie si l'attribut « blocked » est à True et si la variable « iBloque » est égal à 0. Si c'est le cas, on désinstalle le système de « Hooking », on modifie l'attribut « blocked » à False et on ferme le formulaire de « SpyBlock ».

Si on a reçu un message, on envoie au serveur WEB la commande « msgUpdate » avec comme paramètre « message » égal NULL et le N° de session. Ensuite on ouvre une « MessageBox » avec pour contenu le message transmis.

Si on doit effectuer une copie d'écran, on l'effectue avec la méthode « CaptureScreen » de la classe « WindowsDesktop ». On vérifie si le dossier « capture » existe, si ce n'est pas le cas on le crée. Si le nom d'image existe déjà dans ce dossier, on la supprime. Ensuite on sauvegarde l'image dans le dossier « capture ». Ensuite on essaye de se connecter au serveur FTP avec la classe « FTPFactory » pour y envoyer l'image. En cas d'erreur, aucun message n'est affiché. Pour terminer on envoie au serveur WEB la commande « scrUpdate » et comme paramètre l'ID de la copie d'écran et le N° de session.

Nous contrôlons les applications à quitter pour ce faire, on envoie au serveur WEB la commande « appliUpdate » avec pour action « udpKill » et comme paramètre le N° de session. On parcourt la réponse et chaque caractère « ; » indique une nouvelle application à quitter. On essaye de tuer l'application en fonction du N° de processus fourni. En cas d'erreur, rien ne se passe, cela signifie que l'application a déjà été quittée.

Si la « HashTable » des applications reçues (« htAppliRecu ») ne contient pas d'enregistrement, on appelle la méthode « LoadDistantApp ».

On vide la « HashTable » qui contient les titres actifs (« htTitreActif »).

On parcourt les nœuds de l'objet « treeView1 », si le nom du processus se trouve dans la liste des applications interdites (« htAppliInte ») et si ce même

---

<sup>1</sup> Le hooking est un système qui permet de récupérer les actions du clavier même si on ne se trouve pas sur la fenêtre qui exécute le hooking, ce système permet également d'empêcher l'exécution de touches telles que tabulation, windows, control,... Il permet également l'interdiction d'accès à la barre des tâches.

processus ne se trouve pas dans la liste des applications envoyées (« htAppliEnvoye »), on récupère l'ID de l'application avec la méthode « getldByName » et on l'ajoute dans la collection de l'attribut « htAppliEnvoye ». Si le nom du processus figure uniquement dans l'objet « htAppliInte », on le recherche au moyen de la méthode « GetProcessByName » de la classe « Process », si on a un résultat positif, on tue le processus. Pour terminer, on envoie au serveur WEB la commande « appliUpdate » avec l'action « kill » et comme paramètres l'ID de l'application et le N° de session. Si ce processus ne figure pas dans la collection de l'objet « htAppliInte », on vérifie s'il figure dans « htAppliRecu » et si c'est le cas on récupère l'ID de l'application au moyen de la méthode « getldByName », ensuite on contrôle si le nom du processus ne figure dans « htAppliEnvoye » si ce n'est pas le cas on contrôle si l'ID de l'application figure dans « htAppliEnvoye » si ce n'est pas le cas on ajoute le processus dans la collection de « htAppliEnvoye ». Si « htAppliRecu » ne contient pas le nom du processus, on recharge cette liste avec la méthode « LoadDistantApp ». Ensuite, si « htAppliRecu » ne contient toujours pas le nom de l'application, on envoie au serveur la commande « appliUpdate » avec l'action « add » et comme paramètre le nom du processus. En réponse, le serveur WEB nous retourne l'ID de l'application et nous ajoutons dans « htAppliRecu » le nom du processus et l'ID de l'application. On contrôle si « htAppliEnvoye » contient l'ID de l'application, si ce n'est pas le cas on ajoute le nom du processus et l'ID de l'application dans la collection de l'objet « htAppliEnvoye ». Si « htAppliRecu » contient le nom du processus, on récupère l'ID de l'application avec la méthode « getldByName » et on vérifie si « htAppliEnvoye » contient l'ID de l'application, si ce n'est pas le cas on ajoute l'ID de l'application et le nom du processus dans « htAppliEnvoye ». Toujours dans le parcours des nœuds du « treeView1 », on parcourt également les éventuels « sous-nœuds » du nœud actuel dans cette boucle, on supprime les premiers caractères (en partant de la position 0 jusqu'au premier espace) du titre de la fenêtre, on supprime également les espaces se trouvant avant et après le titre de la fenêtre. Si la collection des titres envoyés (« htTitreEnvoye ») ne contient pas le titre de la fenêtre, on concatène à la variable « strArguments » l'ID de l'application (récupéré précédemment) puis le titre de la fenêtre (séparé par le caractère ' ! '). Ensuite on récupère les propriétés du processus actuel avec la méthode « GetProcessesByName » pour que l'on puisse ajouter l'ID du processus et le titre de la fenêtre dans la « htTitreActif ».

Si de nouveaux titres sont à envoyer on les transmet au serveur WEB, ensuite on parcourt la réponse reçue (sous la forme : ID du titre ! Nom de la fenêtre ;) si l'ID du titre est différent de 0, on vérifie si « htTitreEnvoye » contient cet ID, si ce n'est pas le cas on y ajoute l'ID du titre et le titre de la fenêtre. Ensuite on parcourt tous les processus actifs sur la machine et si le titre de fenêtre actuellement lu est égal au titre de fenêtre reçu par le serveur WEB, la variable « ildProc » reçoit l'ID du processus et la variable « dtProc » de type « DateTime » reçoit l'heure de démarrage du processus. Après ces deux boucles, on envoie au serveur WEB la commande « appliUpdate » avec l'action « udp » et comme paramètre les IDs et titres de fenêtre séparés par des ' ! '.

On parcourt la liste des titres actifs (« htTitreActif ») et imbriqué dans cette boucle, on parcourt la liste des titres envoyés (« htTitreEnvoye »), si la collection de l'objet « htTitreActif » ne contient pas la valeur parcourue de « htTitreEnvoye » on concatène à la variable « strArgPro » l'ID du titre qui a été arrêté, on ajoute également cet ID dans le tableau d'entier « aildTitre ». Si, suite à ces deux boucles imbriquées, il y a des titres à transmettre, on envoie au serveur WEB la commande

« appliUpdate » avec l'action « stopPro » et comme paramètres les IDs des titres (séparés par des ' ; ') ainsi que le N° de session.

On parcourt la liste des titres arrêtés (« aildTitre ») et si l'ID est différent de 0, on le retire de « htTitreEnvoye ».

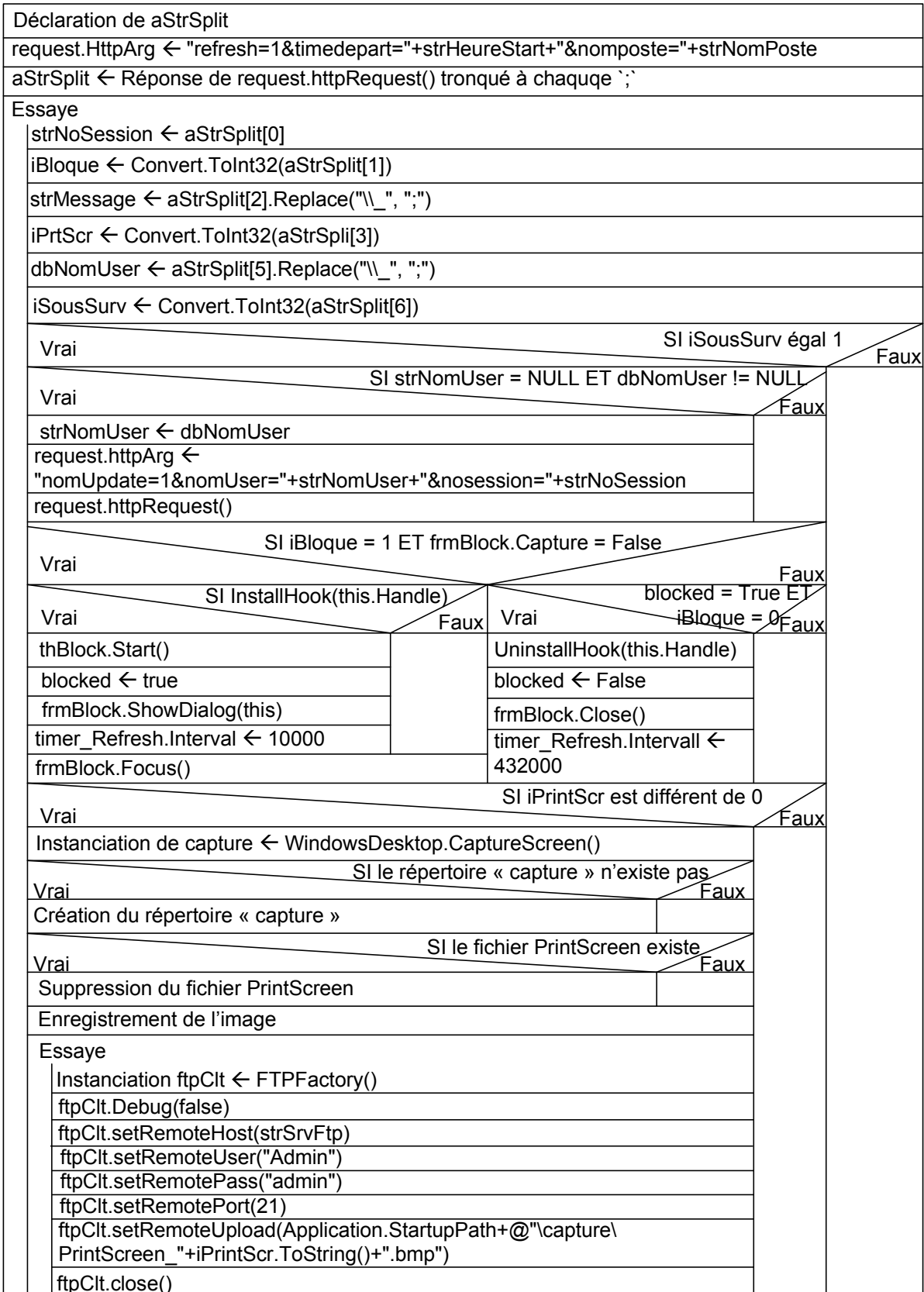
Pour terminer, on envoie au serveur la commande « dateUpdate » avec pour paramètres l'heure actuelle formatée avec la méthode « GetDateTime » et le N° de session.

On remet à 0 l'attribut « iPlante ».

En cas d'erreur dans les précédentes étapes, si « iPlante » est égal à 5 on arrête le « tick » de l'objet « timer\_Refresh », on ouvre le formulaire « setSrv » et on essaye de récupérer les clés de registre « cSpyWeb » et « cSpyFtp », on met à jour la propriété « fullUrl » de l'objet « request » de la classe « ReqHttp » et on réinitialise la valeur de l'attribut « iPlante » à 0.

Si « iPlante » est plus petit que 5 alors on l'incrément de 1.

## Application WEB CIFOM-Spy



**Figure 20 : Structogramme évènement timer\_Refresh\_Tick (1<sup>ère</sup> partie)**

Application WEB CIFOM-Spy

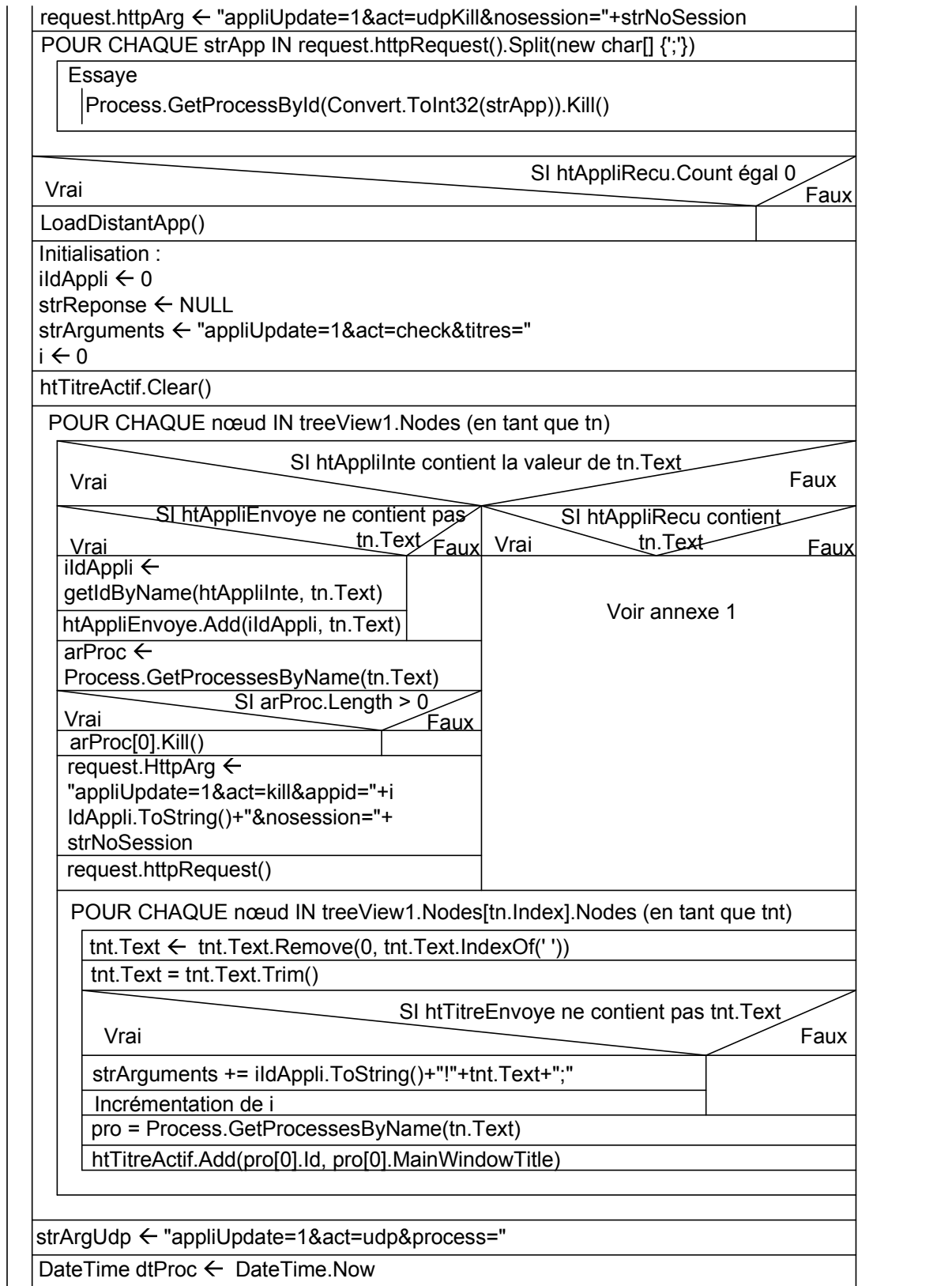


Figure 21 : Structogramme de l'évènement timer\_Refresh\_Tick (2ème partie)

Application WEB CIFOM-Spy

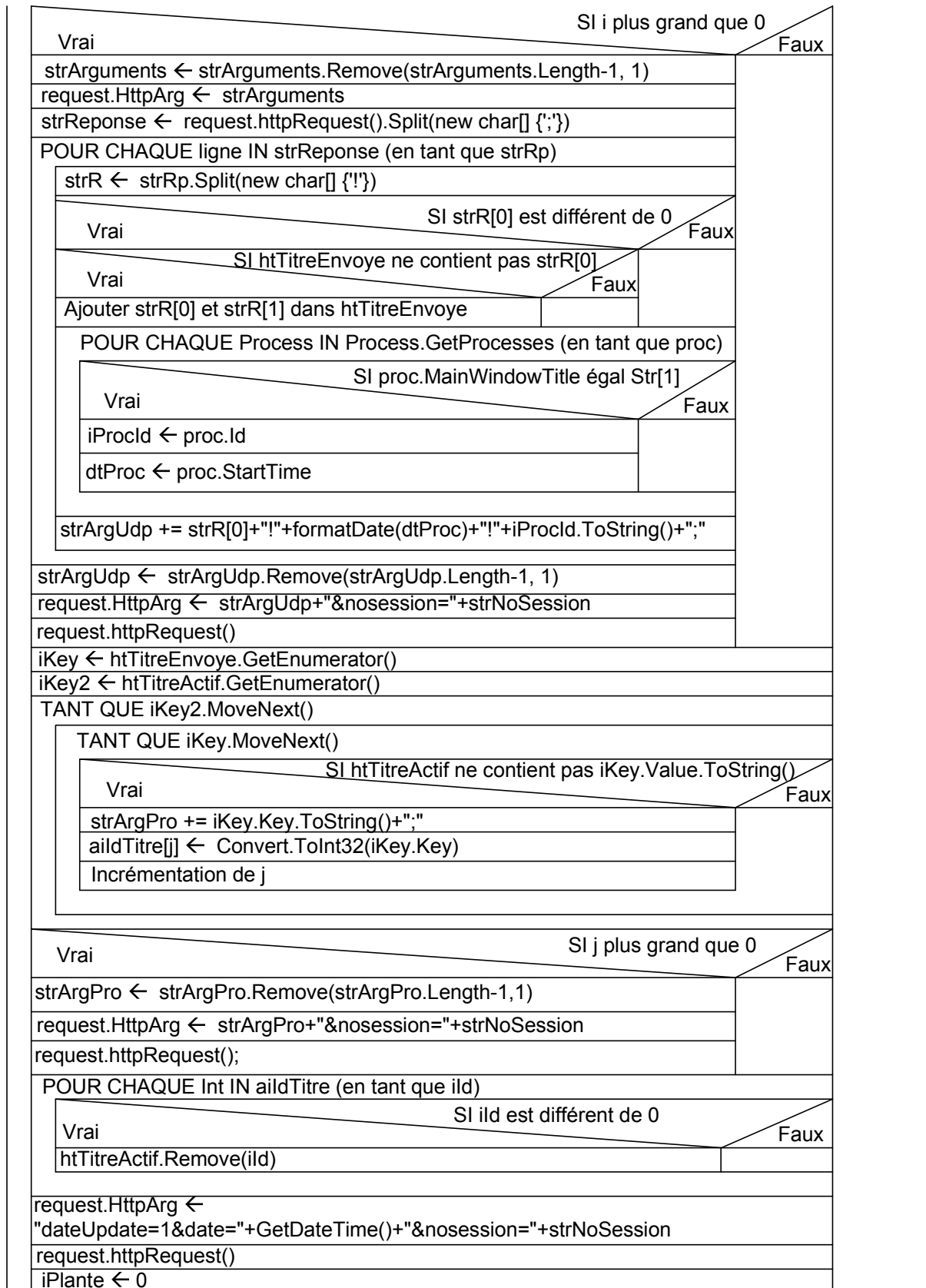


Figure 22 : Structogramme de l'évènement timer\_Refresh\_Tick (3ème partie)

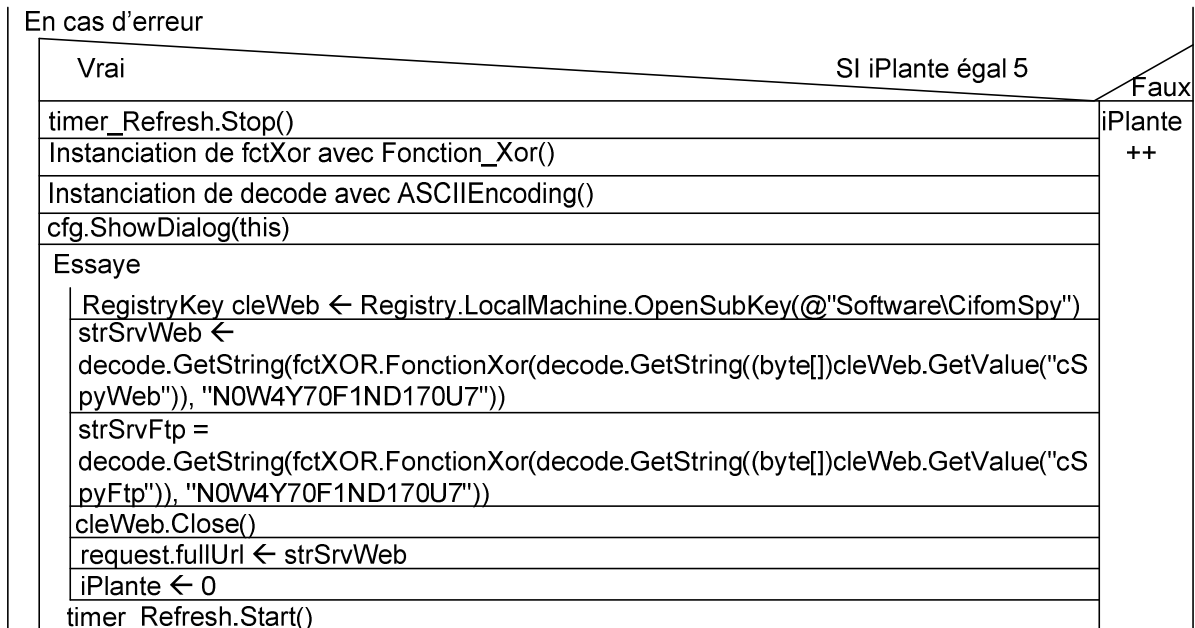


Figure 23 : Structogramme de l'évènement timer\_Refresh\_Tick (4ème partie)

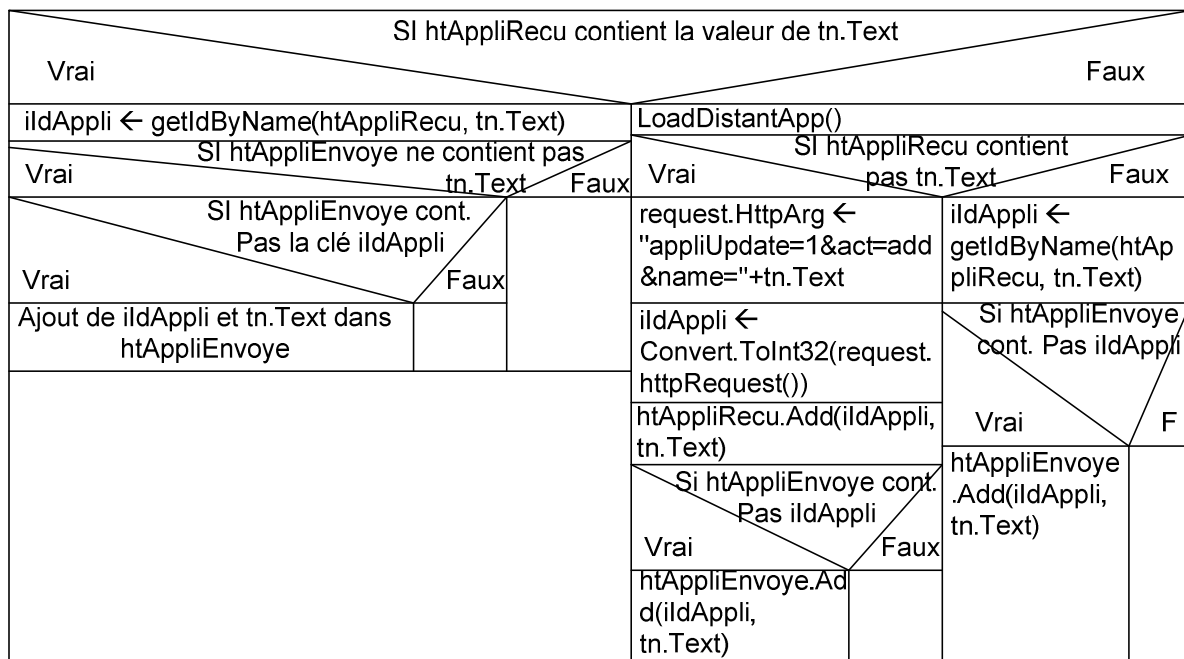


Figure 24 : Annexe 1

treeView1\_AfterSelect

Essaye de récupérer l'ID du processus sélectionné.

S'il n'y parvient pas, efface les divers champs du bas du formulaire.

S'il parvient, récupère les informations du processus sélectionné (ID, application, titre de la fenêtre, heure de départ du processus). Un calcul est fait avec l'heure courante pour indiquer la durée d'ouverture du processus plutôt que son heure d'ouverture.

**Liste des méthodes :**

Méthode	Description
formatDate	Formate une date de type « DateTime » et retourne le résultat en type « string » sous la forme : YYYY-MM-JJ HH :MM :SS

GetDateTime	Récupère l'heure et la transforme en type « string » et formate la chaîne sous la forme : YYYY-MM-JJ HH :MM :SS
getByIdByName	Recherche une clé dans un tableau « HashTable » en fonction de sa valeur et la retourne.
LoadCurrentProcesses	Charge la liste des noms des processus lors du démarrage de l'application.
LoadDistantApp	Récupère les applications enregistrées sur le serveur MySQL.
noManager	Lorsque l'écran est bloqué, un Thread vérifie si le gestionnaire des tâches est lancé, si oui, il l'arrête.

### Explication des méthodes :

#### formatDate

Déclare et initialise les variables de temps (strAnnee, strMois, strJour, strHeure, strMinute, strSeconde) avec un type « DateTime » donné.

Si le contenu des variables de temps (excepté l'année), sont plus court que deux caractères, on ajoute le caractère '0' au début.

On retourne le contenu sous la forme : « YYYY-MM-JJ HH :MM :SS ».

#### GetDateTime

Identique à la méthode « formatDate » mais récupère la date et l'heure actuelle.

#### getByIdByName

Parcourt tout un HashTable donné et tant que l'on n'est pas sur la dernière clé, on vérifie si la clé actuelle est équivalente à une chaîne donnée.

#### LoadCurrentProcesses

LoadCurrentProcesses commence par récupérer la liste des processus actuellement lancés. Elle compte combien il y en a et se sert de ce nombre pour instancier l'attribut astrDebut (tableau de string) avec le nombre exact de cellules (rappel: les tableaux ont une taille fixe et doivent donc être instanciés à la bonne taille).

Ensuite, LoadCurrentProcesses inscrit le nom de tous les processus chargés dans les cellules de astrDebut, à l'exception de Internet Explorer et FireFox, qui doivent toujours être surveillés. Il faut se souvenir qu'ensuite, toutes les applications dont le nom se trouve dans astrDebut ne seront pas surveillées.

(Description et méthode conçus par le supérieur professionnel Mr. Jeanmaire Alain)

#### LoadDistantApp

Envoi au serveur WEB la commande « appliUpdate » avec l'action « recup ».

On vide les HashTable « htAppliInte » et « htAppliRecu ».

On parcourt la réponse et si la valeur n'est pas un « string » vide on récupère l'ID du processus. Si la deuxième partie de la réponse n'est pas égale à 0 on ajoute dans htAppliInte l'ID du processus et le nom de l'application, sinon, on ajoute ceux-ci dans htAppliRecu.

#### noManager

Cette méthode est exécutée par le Thread « thBlock » lorsque l'utilisateur est bloqué.

Tant que l'attribut « blocked » est égale à True, on essaye de récupérer le processus nommé « taskmgr », si on le trouve, on le tue.

### Classe SingleInstanceApp (Form1.cs)

Cette classe hérite de l'Interface « IDisposable » qui permet de libérer les ressources allouées.

#### Liste des méthodes et propriétés utilisées :

Classe	Méthodes/Propriétés	Description
Mutex (System.Threading)	ReleaseMutex	Libération du Mutex.
	WaitOne	Bloque le thread courant jusqu'à réception du signal « Threading.WaitHandle ».

#### Liste des méthodes :

Méthode	Description
SingleInstanceApp	Constructeur.
IsRunning	Détermine si l'application est déjà exécutée ou pas.
Dispose	Libère le Mutex si celui-ci a été acquis.

#### Explications des méthodes :

##### SingleInstanceApp :

Instanciation de l'attribut « \_siMutex » avec « Mutex ».

Initialisation de « \_siMutexOwned » à False.

##### IsRunning

L'attribut « \_siMutexOwned » reçoit \_siMutex.WaitOne(0, true).

Retourne la valeur de « \_siMutexOwned » inversé.

##### Dispose

Si « \_siMutexOwned » est a True, on libère le mutex acquis.

### Classe spyBlock (SpyBlock.cs)

#### Liste des méthodes et propriétés utilisées :

Classe	Méthodes/Propriétés	Description
Screen	PrimaryScreen.WorkingArea.Height	Récupère la hauteur de la résolution de l'écran.
	PrimaryScreen.WorkingArea.Width	Récupère la largeur de la résolution de l'écran.
spyBlock	Height	Hauteur de la fenêtre.
	Width	Largeur de la fenêtre.

#### Liste des évènements :

Evénements	Description
WinForm1_Load	Chargement du formulaire.

**Explications des évènements :**

WinForm1\_Load :

spyBlock.Height reçoit la hauteur de l'écran.

spyBlock.Width reçoit la largeur de l'écran.

**Classe setSrv (setSrv.cs)**

**Liste des méthodes et propriétés utilisées :**

Classe	Méthodes/Propriétés	Description
TextBox	Text	Modifie le texte de la TextBox.
String	Substring	Récupère une partie d'un texte.
	IndexOf	Renvoie la position de la première occurrence d'un/d'une suite de caractères.
	Length	Renvoie le nombre de caractère d'un texte.
MessageBox	Show	Ouvre une fenêtre de type message d'avertissements, informations, ...
Fonction_Xor	FonctionXor	Crypte une chaîne de caractère en XOR.
RegistryKey	Close	Ecris les modifications sur le disque.
	LocalMachine.OpenSubKey	Retourne une sous-clé (lecture seule).
	GetValue	Retourne la valeur d'une clé.
	SetValue	Définit/Modifie la valeur d'une clé.

**Liste des évènements :**

Evènements	Description
button1_Click	Lorsque l'utilisateur clique sur le bouton « Modifier ».

**Explications des évènements :**

button1\_Click :

Récupération des données entrées par l'utilisateur et formatage de celles-ci.

Si le mot de passe est incorrect, on ouvre une MessageBox indiquant qu'il est faux.

Sinon, on supprime les éventuelles clés de registre restantes et on le crée à nouveau avec les nouvelles données.

On ferme le formulaire.

**Classe Fonction\_Xor (Fonction\_Xor.cs)**

**Liste des méthodes et propriétés utilisées :**

Classe	Méthodes/Propriétés	Description
ASCIIEncoding	GetBytes	Convertit une chaîne de type « <i>string</i> » en « <i>byte[]</i> »
	GetString	Convertit une chaîne de type « <i>byte[]</i> » en « <i>string</i> »
Exception	Message	Message en cas d'erreur.
MessageBox	Show	Ouvre une fenêtre de type avertissements, informations,...
MessageBoxButtons	OK	Ajoute un bouton OK à la MessageBox.
MessageBoxIcons	Error	Ajoute un icône Erreur à la MessageBox.
TextBox	Text	Modifie le texte de la TextBox.
XORMelangeur	scramble	Crypte la chaîne.

**Liste des méthodes :**

Méthode	Description
Fonction_Xor	Constructeur.
FonctionXor	Crypte une chaîne de caractère de type « <i>string</i> » en XOR en fonction d'une clé donné.

**Classe FTPFactory (reqFTP.cs)**

**Liste des méthodes et propriétés utilisées :**

Classe	Méthodes/Propriétés	Description
Array	Length	Nombres d'enregistrements dans un tableau
ASCIIEncoding	GetBytes	Convertit une chaîne de type « <i>string</i> » en « <i>byte[]</i> »
	GetString	Convertit une chaîne de type « <i>byte[]</i> » en « <i>string</i> »
Char	IsDigit	Détermine si le caractère est un nombre décimal.
	Parse	Convertit un caractère d'une chaîne « <i>string</i> » en « <i>char</i> ».
Console	WriteLine	Écrit un texte dans une fenêtre DOS et effectue un retour à la ligne.
Dns	Resolve	Résout un nom NetBIOS ou une adresse IP en une instance « <i>IPEndPoint</i> ».

File (System.IO)	Create	Crée un fichier.
	Exists	Vérifie si un fichier existe.
FileStream (System.IO)	Close	Ferme le stream.
	Length	Longueur du fichier (en byte).
	Read	Lit dans le stream et écrit dans le buffer.
	Seek	Déplace le pointeur de lecture.
	Send	Envois le buffer sur la socket.
	Write	Ecrit dans le stream.
Int32	Length	Compte le nombre de caractères.
	Parse	Convertit une chaîne d'un nombre entier de type « string » en « int64 ».
Int64	Parse	Convertit une chaîne d'un nombre entier de type « string » en « int64 ».
Path (System.IO.Path)	GetFileName	Retourne le nom et l'extension d'un fichier.
Socket	Close	Ferme la connexion.
	Connect	Etablit une connexion.
	Connected	Contrôle si la socket est connectée.
	Receive	Lit le flux.
Stream	Close	Ferme le stream.
String	Equals	Compare deux chaînes de caractères.
	IndexOf	Renvoie la position de la première occurrence d'un/d'une suite de caractères.
	Split	Retourne un tableau dont un texte a été séparé en fonction d'un caractère.
	Substring	Récupère une partie d'un texte.

**Liste des méthodes :**

Méthodes	Description
FTPFactory	Constructeur
setRemoteHost	Définis la valeur de l'attribut « remoteHost ».
getRemoteHost	Retourne la valeur de l'attribut « remoteHost ».
setRemotePort	Définis la valeur de l'attribut « remotePort ».
getRemotPort	Retourne la valeur de l'attribut

	« remotePort ».
setRemotePath	Définis la valeur de l'attribut « remotePath ».
getRemotePath	Retourne la valeur de l'attribut « remotePath ».
setRemoteUser	Définis la valeur de l'attribut « remoteUser ».
setRemotePass	Définis la valeur de l'attribut « remotePass ».
getFileSize	Retourne la taille d'un fichier.
login	Connexion au serveur FTP.
setBinaryMode	Passes en mode binaire lors d'un téléchargement.
upload (surchargé)	Envoi un fichier.
chdir	Change le répertoire courant.
close	Ferme la connexion.
setDebug	Passes en mode debug.
readReply	Lit une réponse
cleanup	Vide les ressources.
readLine	Lit une seule ligne de la réponse du serveur FTP.
sendCommand	Envoi une commande au serveur FTP.
createDataSocket	Crée un socket de connexion au serveur FTP.

**Explications des méthodes :**

FTPFactory

Instancie les attributs à leurs valeurs par défaut.

setRemoteHost

Change la valeur de « remoteHost ».

getRemoteHost

Retourne la valeur de « remoteHost ».

setRemotePort

Modifie la valeur de l'attribut « remotePort ».

getRemotePort

Retourne la valeur de l'attribut « remotePort ».

setRemotePath

Modifie la valeur de l'attribut « remotePath ».

getRemotePath

Retourne la valeur de l'attribut « remotePath ».

setRemoteUser

Modifie la valeur de l'attribut « remoteUser ».

setRemotePass

Modifie la valeur de l'attribut « remotePass ».

#### getFileSize

Si on n'est pas connecté, on effectue la connexion.

On envoie la commande « SIZE ».

Si la valeur retournée est égale à 213, on met la taille du fichier dans la variable « size ».

Sinon on génère une exception.

On retourne la variable « size ».

#### login

Création du socket.

On essaye de se connecter, sinon on génère une exception.

On lit la réponse du socket.

Si la valeur retournée n'est pas égale à 220, on ferme la connexion et on génère une exception.

Si on est en mode debug, on écrit dans la console « USER » + « remoteUser ».

On envoie la commande « USER » au serveur.

Si la valeur retournée n'est pas égale à 230 ou 331, on libère les ressources et on génère une exception.

Si la valeur n'est pas égale à 230, si on est en mode debug, on écrit dans la console « PASS xxx ». On envoie la commande « PASS » au serveur FTP, si la valeur retournée n'est pas égale à 230 ou 202, on libère les ressources et on génère une exception.

On définit l'attribut « logged » à True.

On écrit dans la console « Connecte to ».

On change de répertoire.

#### setBinaryMode

Si « mode » est égal à True, on envoie la commande « TYPE I » sinon on envoie la commande « TYPE A ».

Si la valeur retournée n'est pas égale à 200, on génère une exception.

#### upload

On appelle la méthode surchargée upload.

#### upload (string, bool)

Si on n'est pas connecté, on se connecte.

Création du socket.

Si le flag « resume » est activé, on essaye de passer en mode binaire et on récupère la taille du fichier à envoyer. En cas d'erreur, on définit la variable « offset » à 0.

Si « offset » est plus grand que 0, on envoie la commande « REST », si la valeur retournée n'est pas égale à 350, on définit « offset » à 0.

On envoie la commande « STOR », si la valeur retournée n'est pas égale à 150 ou à 125, on génère une exception.

Instanciation d'un « FileStream ».

Si « offset » est différent de 0, si « debug » est à True, on écrit dans la console « seeking to ». On déplace le pointeur de lecture du fichier à la valeur de « offset ».

On écrit dans la console « Uploading file to ».

Tant qu'il y'a des données à lire, on envoie ces « bytes » dans sur le socket.

On ferme le fichier.

On lit la réponse.

Si la valeur retournée n'est pas égale à 226 ou 250, on génère une exception.

#### chdir

Si le nom du répertoire passé en paramètre est équivalent à `.` , on ne fait rien de plus.

Si on n'est pas connecté, on se connecte.

On envoi la commande « CWD ».

Si la valeur retournée n'est pas égale à 250, on génère une exception.

On change la valeur de « remotePath ».

On écrit dans la console « Current directory is ».

#### close

Si le socket n'est pas égal à NULL, on envoi la commande « QUIT ».

On vide les ressources et on écrit dans la console « Closing... ».

#### setDebug

Modifie la valeur de l'attribut « debug ».

#### readReply

« reply » égal la réponse de la méthode « readLine ».

retValue égal la conversion des trois premières caractères de « reply » convertit en « Int ».

#### cleanup

Si le socket n'est pas NULL, on ferme le socket et on lui donne la valeur NULL.

On met l'attribut « logined » à False.

#### readLine

Tant que c'est vrai, on lit la réponse disponible sur le socket que l'on convertit en « string ». Si « bytes » est plus petit que la taille de « buffer », on quitte la boucle.

On retourne le résultat complet de la réponse FTP.

#### sendCommand

On convertit en « tableau de Byte » la valeur passée en paramètre et on envoi la valeur sur le socket.

On lit la réponse.

#### createDataSocket

On envoi la commande « PASV ».

Si la valeur retournée n'est pas égale à 227, on génère une exception.

Pour i allant de 0 jusqu'au nombre de caractères ET tant que « partcount » est plus petit ou égal à 6. Si « ch » est un nombre, on concatène « ch » à « buf ». Sinon, si « ch » n'est pas égale à ` , ` on génère une exception.

Si « ch » est égal à ` , ` ou si i + 1 égal « len », on essaye de récupérer une partie de la réponse, en cas d'échec on génère une exception.

Ensuite, on connecte le socket.

### **Classe ReqHttp (ReqHttp.cs)**

<b>Classe</b>	<b>Méthodes/Propriétés</b>	<b>Description</b>
---------------	----------------------------	--------------------

Application WEB CIFOM-Spy

Array	Length	Nombres d'enregistrements dans un tableau
Encoding (System.Text)	GetBytes	Encode un texte de type « string » en un tableau de bytes.
	GetEncoding	Retourne un type « Encoding ».
HttpUtility	UrlEncode	Convertit un tableau de bytes en string pour une transaction http.
Stream	Close	Ferme le stream.
	Write	Ecrit sur le stream.
StreamReader	Close	Ferme le stream.
	Read	Compte le nombre de caractère dans le buffer d'un stream.
String	EndWith	Contrôle si un texte se termine par une chaîne donnée.
	Insert	Insert une chaîne de caractères dans un texte.
	Length	Renvoie le nombre de caractère d'un texte.
	Remove	Supprime une partie du texte.
	StartWith	Contrôle si un texte commence par une chaîne donnée.
StringBuilder	Append	Ajoute un tableau de caractères à la fin du « StringBuilder ».
WebRequest	ContentLength	Longueur des données qui vont être envoyées.
	ContentType	Type du contenu à envoyer.
	Create	Initialise une nouvelle instance pour une URI donné.
	Method	Définis quel méthode utilisé (GET ou POST).
	GetRequestStream	Récupère le stream sur lequel envoyer les données.
	GetResponse	Récupère la réponse d'une précédente requête.
	GetResponseStream	Récupère le stream sur lequel la réponse va être reçue.

PreAuthenticate	Indique si faut une identification est requise.
-----------------	---

**Liste des méthodes :**

Méthodes	Description
ReqHttp	Constructeur
HttpRequest	Effectue une requête HTTP.

**Description des méthodes :**

HttpRequest

Instanciation de « encode », « req ».

On passe le type de donnée à envoyer en POST.

On définit le type de contenu à « application/x-www-form-urlencoded ».

Instanciation de « UriEncoded ».

Si il y'a des arguments à envoyer on les parcourt, les encode et les ajoute à l'objet UriEncoded.

On convertit l'URL en tableau de « byte ».

On essaye d'envoyer les données, en cas d'erreur, on retourne « Serveur indisponible. Réessayer plus tard. ».

On essaye de recevoir une réponse, en cas d'erreur, on retourne « Erreur lors de la réception de donnée. Réessayer plus tard. ».

Une fois la réponse reçue, on lit le stream, le résultat est ajout à « strRes ».

On ferme la connexion.

On retourne le « strRes ».

**Classe WindowsDesktop (WindowsDesktop.cs)**

**Liste des méthodes et propriétés utilisées :**

Classe	Méthodes/Propriétés	Description
Image (System.Drawing)	FromHbitmap	Crée une image.

**Liste des méthodes :**

Méthodes	Description
CaptureScreen	Effectue une copie d'écran.

**Description des méthodes :**

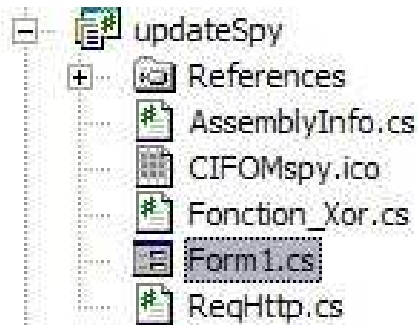
CaptureScreen

Instancie quatre pointeurs.

Effectue la copie d'écran et la retourne sous forme d'image.

## Application updateSpy

### Carte de l'application



Dossier racine.

Dossier des références de l'application..

- Information d'assemblage.

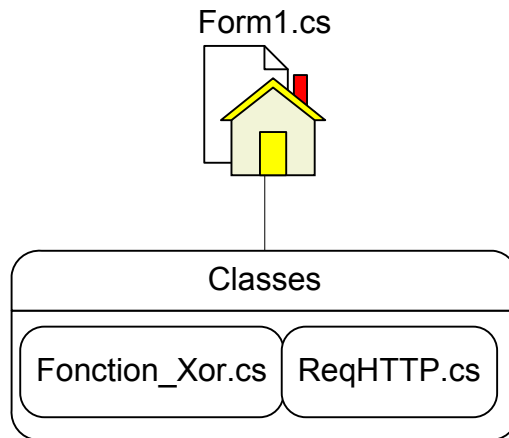
- Icône de l'application.

- Classe « FonctionXOR ».

- Formulaire principale.

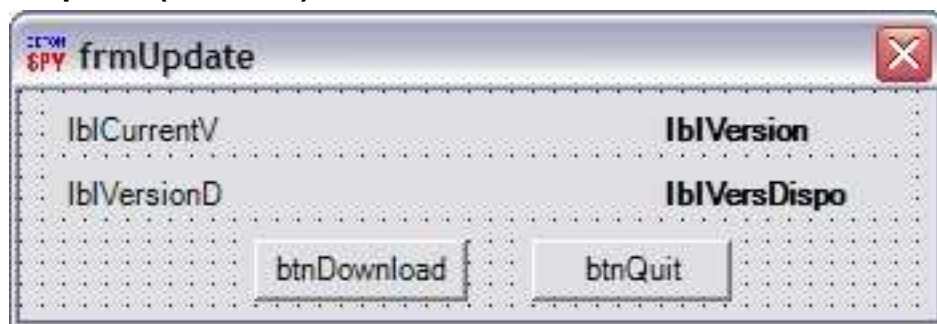
- Classe « ReqHTTP ».

### Schéma Synoptique



### Maquettes

#### frmUpdate (form1.cs)



### Classes, méthodes et propriétés utilisées

#### Classe frmUpdate (form1.cs)

##### Liste des méthodes et propriétés utilisées

Classe	Méthodes/Propriétés	Description
Application	Exit	Quitter l'application
	Run	Démarre l'application
Array	Length	Nombres

Application WEB CIFOM-Spy

		d'enregistrements dans un tableau
ASCIIEncoding	GetString	Convertit une chaîne de type « <i>byte[]</i> » en « <i>string</i> »
Environment	DoEvents	Exécute toutes les actions en attente, dans la queue.
	MachineName	Nom NetBIOS de la machine.
	StartupPath	Chemin d'exécution de l'application.
	ProductVersion	Retourne le n° de version de l'application.
Fonction_Xor	FonctionXor	Crypte une chaîne de caractère en XOR.
Label	Text	Modifie le texte d'un objet « label ».
MessageBox	Show	Ouvre une fenêtre de type message d'avertissements, informations, ....
Process (System.Diagnostics)	GetProcessesByName	Récupère tous les process selon un nom.
	Kill	Force le processus à s'arrêter.
	Start	Démarré un processus.
	WaitForExit	Met le Thread courant en pause et le relance dès le processus arrêté.
RegistryKey	Close	Ecris les modifications sur le disque.
	LocalMachine.OpenSubKey	Retourne une sous-clé (lecture seule).
	GetValue	Retourne la valeur d'une clé.
ReqHttp	FullUrl	Adresse complète de la page à laquelle envoyer les données.
	HttpArg	Arguments à envoyer.
	HttpRequest	Envoi les données.
String	Split	Retourne un tableau dont un texte a été séparé en fonction d'un caractère.
WebClient	DownloadFile	Télécharge un fichier distant sur le disque local.
	Dispose	Libère toutes les ressources utilisées par WebClient.
WinForm	Close	Ferme le formulaire.

### Liste des évènements

btnDownload_Click	Lorsque l'utilisateur clique sur le bouton « Télécharger ».
btnQuit_Click	Lorsque l'utilisateur clique sur le bouton « Quitter ».
Form1_Load	Au chargement de l'application.

### Explications des évènements

#### btnDownload\_Click

On cherche le processus « cifomspy » et on le quitte, ensuite on télécharge les fichiers à télécharger, puis on relance le processus « cifomspy » et on quitte l'updater.

#### btnQuit\_Click

On quitte l'updater sans rien faire.

#### Form1\_Load

On appelle la méthode « GetServeurWeb », si elle retourne FALSE, on affiche un message box et on quitte l'application. Si elle place le flag « doUpdate » à TRUE, on affiche un message box demandant si on désire télécharger l'application, sinon on quitte l'application.

### Liste des méthodes

GetServeurWeb	Récupère l'adresse du serveur WEB et fait la demande si il y a une mise à jour.
---------------	---

### Explications des méthodes

#### GetServeurWeb

Pour commencer, il faut essayer de récupérer l'adresse du serveur WEB dans la base de registre, en cas d'échec on retourne FALSE.

Lorsque l'on dispose de l'adresse du serveur WEB, on envoie la demande de mise à jour à la page « spy/update.inc.php ».

Ensuite, lors de la lecture du résultat, on met le flag « doUpdate » à TRUE s'il y a des fichiers à mettre à jour. Sinon, on le met à FALSE.

#### **Classe Fonction\_Xor (Fonction\_Xor.cs)**

(Voir classe Fonction\_Xor de la page 40).

#### **Classe ReqHttp (ReqHttp.cs)**

(Voir classe ReqHttp de la page 45).

## Documentation utilisateur

### *Installation du CIFOM-Master*

#### **Le serveur WEB**

Avant de pouvoir utiliser l'application WEB, il vous faut installer « EasyPHP 1.8 » pour ce faire, téléchargez l'installateur automatique à cette adresse :

<http://www.easyphp.org/telechargements/dn.php?F=easyphp1-8>

Double cliquez sur l'exécutable téléchargé et sélectionnez le répertoire d'installation et suivez les instructions.

#### **Configuration**

Afin de permettre aux autres ordinateurs d'accéder à votre serveur WEB, il y a une petite modification à effectuer dont voici la procédure :

Tout d'abord vérifiez qu'EasyPHP est correctement lancé, si c'est le cas, vous devriez voir apparaître l'icône d'EasyPHP dans le system Tray comme l'image ci-dessous le démontre.



Figure 25 : Exemple d'EasyPHP lancé



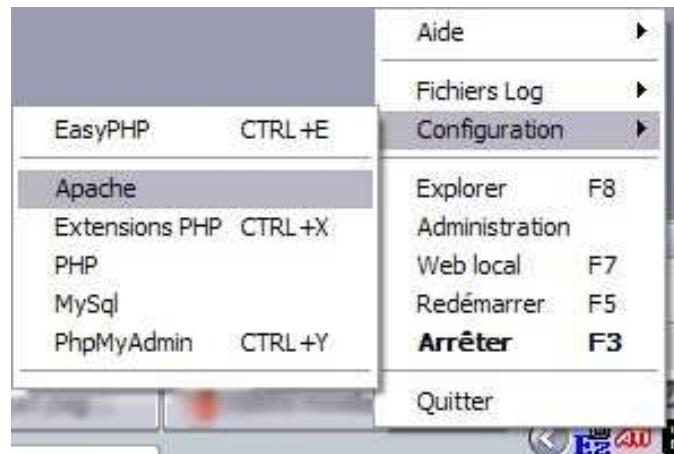


Figure 27 : Atteindre la configuration d'Apache

Après cet étape, le bloc note s'ouvre, dans celui-ci recherchez la ligne « Listen 127.0.0.1:80 » et ajoutez un `#` devant cette ligne. Enregistrez la modification et redémarrez le serveur Apache. Pour redémarrer le serveur Apache, double cliquez sur l'icône EasyPHP du système tray, le formulaire d'EasyPHP s'ouvre sur lequel se trouve deux boutons. Cliquez sur le premier bouton nommé Apache, dans le menu contextuel qui apparaît sélectionnez Arrêter. Une fois que l'image du feu à gauche du bouton Apache passe au rouge, cliquez à nouveau sur le bouton Apache et dans le menu contextuel sélectionnez démarrer.



Figure 28 : Arrêter et Démarrer Apache.

Si on fait de cette manière pour redémarrer Apache et qu'on n'utilise pas le menu Redémarrer comme montré sur les images, c'est parce que celui-ci à du mal à fonctionner correctement.

### Ajout de la base de données

Ouvrez votre navigateur WEB et entrez pour URL : <http://localhost/mysql/> une fois sur la page d'accueil de PHPMyAdmin, créez une base de données nommé « cifomspy » lorsque la base est créé, cliquez sur le bouton SQL comme le montre l'image ci-dessous.



Une fois la page chargée, cliquez sur le bouton Parcourir et allez chercher le fichier SQL à la racine du dossier « cifomspy » puis cliquez sur le bouton Exécuter.

### Copie de l'application WEB

Copiez le dossier « cifomspy » qui contient l'application WEB et collez le dans : « C:\Program Files\EasyPHP1-8\www ». Afin de pouvoir tester si la copie c'est déroulée correctement, ouvrez une fenêtre de votre navigateur WEB et entrez l'URL : <http://localhost/cifomspy>

### Le serveur FTP

Afin de pouvoir recevoir les images envoyées par le CIFOM-Spy, il faut installer un serveur FTP disponible à cette adresse : <http://php.typsoft.com/download.php?id=1>  
Une fois l'exécutable téléchargé, exécutez-le et suivez les instructions.

### Configuration

Lorsque le serveur à été installé, vous pouvez le démarrer depuis : Tous les programmes → TYPSoft FTP Server → TYPSoft FTP Server. Lorsqu'il est actif, une icône apparaîtra dans le système tray comme l'image ci-dessous.



Figure 29 : SysTray lorsque FTPServer est en cours d'exécution

Pour atteindre le menu de configuration, double cliquez sur l'icône de FTPServer pour ouvrir la fenêtre. Ensuite aller dans le menu Setup puis sélectionnez FTP.



Figure 30 : Atteindre la configuration du serveur FTP

Une fois la nouvelle fenêtre ouverte, suivez la configuration ci-dessous :

- Cochez Launch FTP Server on Windows Startup
- Cochez Start in SysTray
- Sélectionnez Language français
- Cliquez sur Save puis sur Close
- La langue du programme va changer en se mettre en français



**Figure 31 : Configuration du serveur FTP**

Ensuite, allez dans le menu Configuration et sélectionnez Utilisateurs.

Dans la fenêtre qui s'ouvre, cliquez sur le bouton Nouvel Utilisateur, dans la fenêtre qui s'ouvre, entrez le nom Admin. Ensuite, indiquez comme mot de passe admin.

Cliquez sur le bouton vous permettant d'aller chercher le dossier racine auquel l'utilisateur aura accès (C:\Program Files\EasyPHP1-8\www\cifomspy\img).

Cochez la case Affichage du Répertoire en Mode Virtuel.

Dans la liste de sélection de droite, sélectionnez le dossier qui vient de s'ajouter et sélectionnez les cases à cocher suivantes :

- Upload
- Créer
- Sous-Répertoire Inclus

Cliquez ensuite sur Enregistrer.

Voici à quoi devrait ressembler les champs lorsque la configuration est terminée.

## Installation du CIFOM-Master

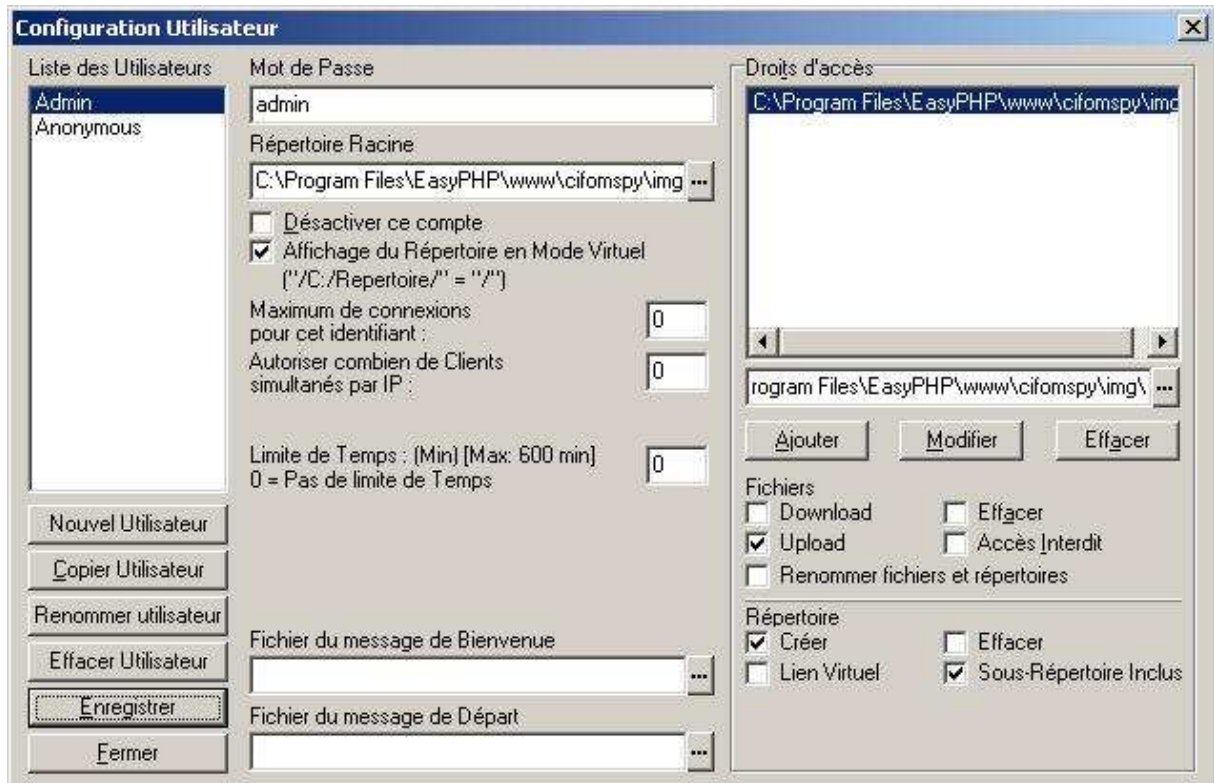


Figure 32 : Configuration de l'utilisateur du serveur FTP

Pour terminer, sélectionnez à gauche, l'utilisateur Anonymous et cochez la case Désactiver ce compte. Cliquez sur Enregistrer et puis sur Fermer.

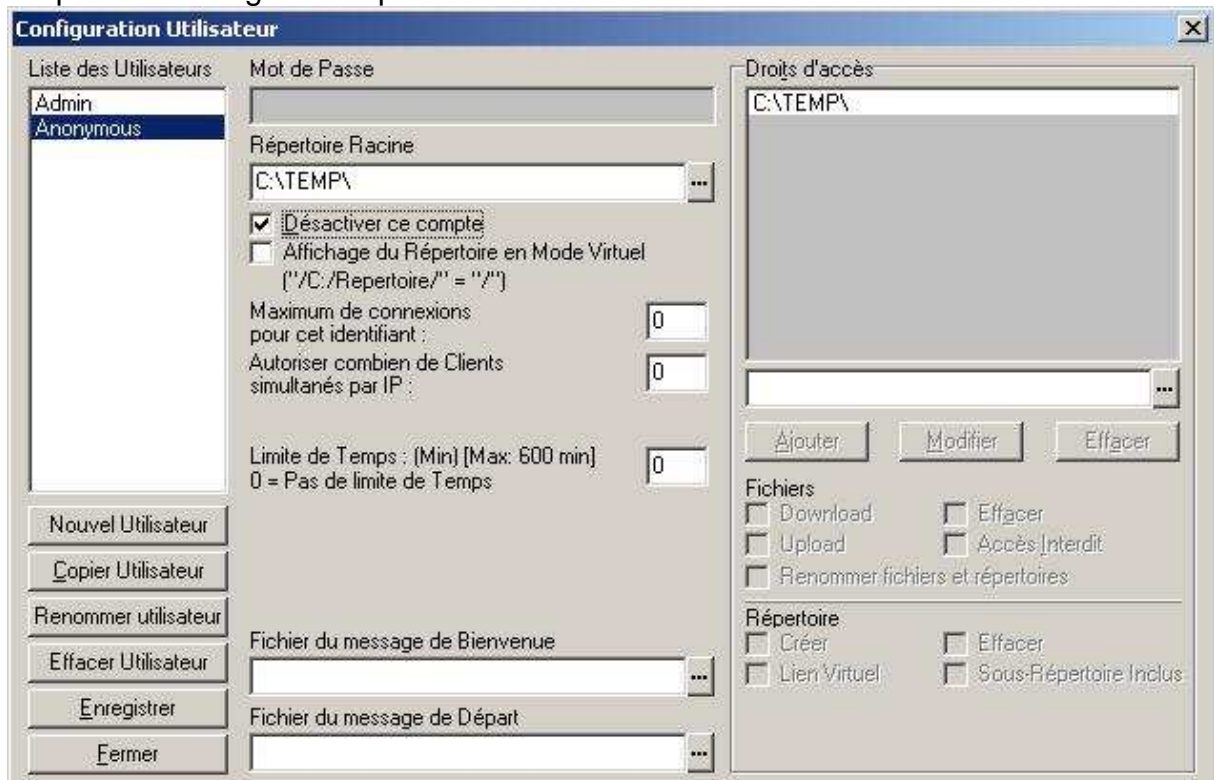


Figure 33 : Désactivation du compte Anonymous

## Utilisation du CIFOM-Master

Lorsque vous arrivez sur la page d'accueil du CIFOM-Master, si vous êtes logué, vous arrivez directement sur la page de surveillance, dans le cas contraire, vous êtes redirigé sur la page de connexion, sur laquelle il vous sera demandé un nom d'utilisateur et un mot de passe ainsi que la page sur laquelle vous voulez aller. Une fois la connexion réalisée, vous êtes redirigé sur la page que vous avez précédemment sélectionnée.

### Page de connexion

La page de connexion, vous permet d'entrer votre nom d'utilisateur et votre mot de passe ainsi que de sélectionner sur quel page vous désirez vous rendre. En cas d'erreur dans le nom d'utilisateur ou le mot de passe, vous revenez sur la page de connexion et un message s'affiche en rouge au dessus des champs du formulaire.

### Page de configuration

La page de configuration vous permet de configurer vos paramètres. Vous avez la possibilité de modifier les informations suivantes :

- Le nom d'utilisateur
- Le mot de passe
- Le N° de salle
- Le nombre de poste dans la salle
- La liste des postes dans la salle
- Le nombre de poste par rangées
- L'alignement des postes par rangées
- Le temps de rafraîchissement de la page de surveillance

Voici un exemple de configuration :

**Formulaire de configuration**

Aller à : [Page de surveillance](#) - [Page de statistiques](#)

Nom d'utilisateur\* : Nic0tiN

Mot de passe :  (Laisser vide si inchangé.)

N° de salle\* : 226

Nombre de poste\* : 14

Liste des postes de la salle\* : 01;02;03;04;05;06;07;08  
(Séparer par des ';' Ex.: 01;02;03)

Nombre de poste par rangées\* : 4;5;5  
(Séparer par des ';' Ex.: 1;2)

Alignement des postes par rangées\* : r;1;1  
(Séparer par des ';' Ex.: 'r','l') l = gauche, r = droite

Temps de rafraîchissement\* : 60 (Rafraîchissement en secondes de la page surveillance)

\* Paramètres obligatoires.

Figure 34 : Exemple de configuration d'un utilisateur du CIFOM-Master

### Page de surveillance

Une fois la configuration de la salle effectuée, vous pouvez vous diriger sur la page de surveillance. Celle-ci vous donne les possibilités suivantes :

- Actualiser l'affichage lors d'une nouvelle session
- Bloquer 1 PC
- Bloquer tous les PC
- Détail d'une tâche
- Requérir une copie d'écran
- Stopper une tâche
- Envoyer un message à 1 PC
- Envoyer un message à tous les PC
- Réinitialiser (ATTENTION : Réinitialise toute les sessions)

Avant d'interagir avec un client, vérifiez que le temps avant la mise à jour soit suffisant, vous pouvez vérifier le temps à droite de la zone de texte en bas de la page.

### Signification des couleurs

Il y a 4 couleurs en fonction du temps de rafraîchissement envoyés par les clients, elles vous sont listées ci-dessous.

Bleu : Aucun problème.



Figure 35 : Le client n'a pas de problème.

Jaune : Le client n'a pas encore donné signe de vie.



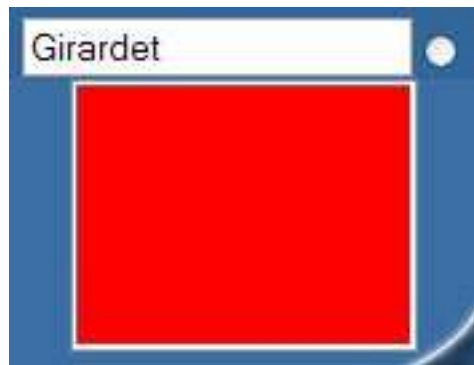
Figure 36 : Poste sous contrôle pas encore démarré.

Orange : Le client a démarré mais son dernier temps de rafraîchissement remonte à entre 5 et 10 minutes.



**Figure 37 : Poste sous surveillance sans réponse depuis 5 minutes**

Rouge : Le client n'a pas répondu depuis 10 minutes ou alors celui-ci a détecté une fraude.



**Figure 38 : Client en fraude potentiel**

### **Actualiser**

Pour mettre un nouveau client sous surveillance, introduisez le nom de l'élève dans le champ texte correspondant au PC et cochez le bouton radio à sa droite. Cliquez ensuite sur le bouton Actualiser et l'interface changera et s'affichera comme suit :



**Figure 39 : Exemple d'un poste mis sous surveillance.**

Si vous voulez simplement actualiser la page avant que le temps ne soit écoulé et sans effectuer aucune action, vous pouvez sélectionner le radio bouton Sans action à droite du bouton Réinitialiser et cliquez sur le bouton réinitialiser.

### **Blocage/Déblocage**

Pour bloquer un client, sélectionner le bouton radio à droite du nom de l'élève et cliquez sur le bouton Bloquer 1 PC. Lorsque le client est bloqué, le champ texte où se trouve le nom de l'élève apparaît en rouge comme ci-dessous :

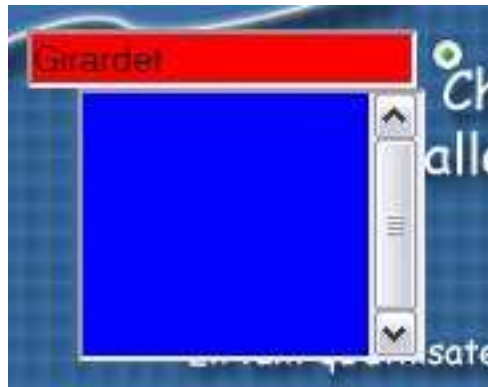


Figure 40 : Exemple d'un poste bloqué.

Pour le débloquent, il vous suffit de faire la même procédure.

Pour bloquer tous les clients, vous n'êtes pas obligé de sélectionner un bouton radio, il vous suffit simplement de cliquer sur le bouton Bloquer TOUS. Une fois tous bloqué, le texte du bouton change et devient Libérer TOUS, sur lequel il vous faudra cliquer lorsque vous voudrez libérer les élèves.

### Détails tâches

Lorsque vous désirez plus de renseignement sur l'application qui s'exécute sur l'ordinateur d'un élève, sélectionnez le bouton radio à droite du nom de l'élève, sélectionnez une application puis cliquez sur le bouton Détails tâches. Le résultat apparaîtra dans la zone de message.



Figure 41 : Exemple d'une tâche détaillée

### Copie d'écran

Au moment où vous désirez voir exactement l'écran de l'élève, vous pouvez sélectionner le bouton radio à droite de son nom et cliquez sur le bouton Copie d'écran.

Lorsque le client aura envoyé l'image sur le serveur, une fenêtre s'ouvrira (pensez à désactiver votre anti-popup). Dans cette fenêtre, vous aurez la copie d'écran ainsi qu'un listing des applications qui sont actuellement lancées et depuis quand elles le sont.

### Stop tâches

Lorsque l'élève exécute une application qu'il ne devrait pas, vous pouvez à tout moment sélectionner le bouton radio à droite du nom de l'élève, sélectionner l'application à arrêter et cliquer sur le bouton Stop tâches.

### Message ciblé

Pour envoyer un message à un élève, écrivez votre message dans la zone de texte en bas de la fenêtre, sélectionnez le bouton radio à droite du nom de l'élève et cliquez sur le bouton Msg ciblé.

### Message collectif

Lorsque tous les élèves doivent recevoir le même message, écrivez votre message dans la zone de texte et cliquez sur le bouton Msg collectif.

### Réinitialiser

Attention, ce bouton a pour effet d'effacer toutes les sessions en cours, c'est-à-dire qu'il va indiquer aux espions qu'une nouvelle journée commence.

### Fraude

Lorsqu'un utilisateur à arrêter puis redémarrer le client, la liste de sélection du client deviendra rouge. Celle-ci peut également rougir lorsque le client CIFOM-Spy ne s'est pas actualisé depuis 5 minutes, cela peut être dû soit à une erreur du client soit parce que l'élève a quitté l'application.



Figure 42 : Exemple d'un utilisateur ayant fraudé

Pour retirer le fond rouge de la liste de sélection, vous devez sélectionner le bouton radio à droite du nom de l'élève, retirer le nom de l'élève, cliquer sur le bouton Actualiser puis remettre le nom de l'élève et à nouveau cliquer sur le bouton Actualiser.

### Page des statistiques

La page des statistiques va vous permettre de savoir quels utilisateurs ont lancés quelles applications et à quelles heures. Elle permet également de savoir lorsqu'un utilisateur a triché ou non. Les enregistrements sont affichés chronologiquement. Vous pouvez naviguer entre les différentes dates (ultérieur ou égale à aujourd'hui) et les différents postes de la salle.

### Mise à jour du CIFOM-Spy

Lorsque vous recompilez le client du CIFOM-Spy, vous allez devoir mettre les fichiers compilés directement dans le dossier « files » du CIFOM-Master. Voici comment procéder :

Dans un premier temps, lors que vous avez compilé la nouvelle version du CIFOM-Spy, copiez l'exécutable de celui-ci et allez le coller dans « c:\Program Files\EasyPHP1-8\www\cifomspy\files ».

Faites un clique droit sur le fichier et sélectionnez « Propriété » et sélectionnez l'onglet « Version », sélectionnez la valeur à droite de « Version du fichier » (qui devrait ressembler à W.X.YYYY.ZZZZ) et utilisez le raccourci « Ctrl+C » pour copier la ligne puis cliquez sur « OK ».

Faites un clique droit sur le fichier « cifomspy.exe » et sélectionnez « Renommer » le nouveau nom devra correspondre à : « cspy\_W.X.YYYY.ZZZZ.exe ».

### Installation du CIFOM-Spy

Pour installer le logiciel CIFOM-Spy, le logiciel .NET Framework doit être installé.

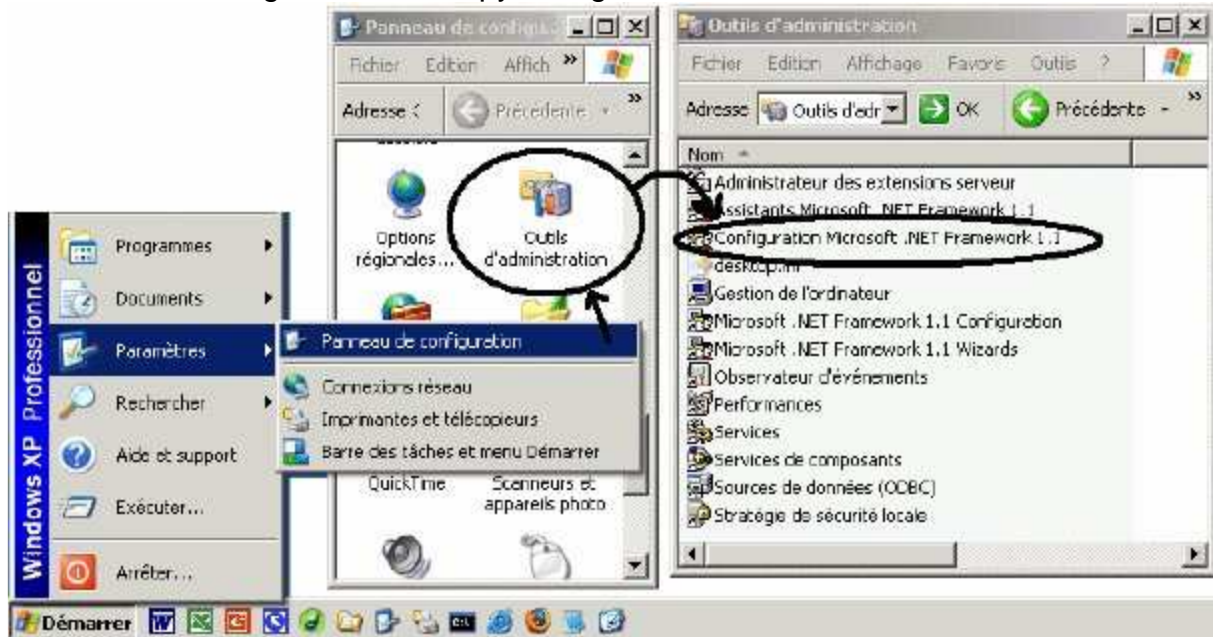


Figure 43 : Contrôler si le framework est installé

Vérifiez que le .NET Framework est installé en cliquant sur Démarrer → Paramètres → Panneau de configuration → Outils d'administration. Si le Framework est installé, il apparaît dans la liste des outils (voir figure ci-dessus) Si ce logiciel (taille : 22MB) n'est pas installé, il peut être téléchargé sur :

<http://www.microsoft.com/france/msdn/netframework/default.aspx>

Double-cliquer sur le fichier téléchargé et suivez les instructions à l'écran.

Pour installer le CIFOM-Spy, copiez l'exécutable cifomspy.exe dans un dossier quelconque avec la DLL qui l'accompagne et le fichier .reg.

Pour modifier le fichier addtorun.reg, faites un clique droit sur celui-ci et sélectionnez Modifier. Le Bloc-Note s'ouvre et vous pouvez modifier le chemin par celui où vous avez mis l'exécutable cifomspy.exe.

## Configuration

Au premier lancement de l'application CIFOM-Spy, le programme vous demande d'introduire les adresses des serveurs WEB, FTP ainsi qu'un mot de passe autorisant la modification. Le mot de passe de configuration est n0w4y70 il est nécessaire de l'entrer correctement et en respectant la casse sinon il vous sera impossible de modifier les paramètres.

La fenêtre de configuration peut s'ouvrir lorsque l'application CIFOM-Spy a un problème et a générée cinq erreurs.

L'adresse du serveur WEB ne se limite pas à <http://serveur.dom>, il faut indiquer l'adresse qui pointe jusqu'au fichier spy/common.inc.php comme le présente l'image ci-dessous.



Figure 44 : Configuration des serveurs

## Mise à jour

Il est possible qu'au lancement du CIFOM-Spy, celui-ci vous demande si vous désirez le mettre à jour. Vous pouvez confirmer en cliquant sur OK ou sur Annuler.

Si vous demandez OK, il vous faudra confirmer votre demande en cliquant sur Télécharger, suite à cela, l'updater se fermera et le client CIFOM-Spy se relancera.

## Programme minimisé

Le logiciel CIFOMspy s'enclenche automatiquement au démarrage de votre ordinateur et affiche une icône dans la barre système près de l'horloge.



Figure 45 : Icône du CIFOM-Spy, près de l'horloge

En passant la souris sur l'icône, une bulle indique qu'il est interdit d'arrêter ce programme.

Vous pouvez afficher la liste des tâches surveillées par CIFOMspy en cliquant avec le bouton droit de la souris. Le menu qui s'ouvrira vous permet d'afficher la version de CIFOM SPY et d'ouvrir la fenêtre de suivi des tâches surveillées.

(Description faites par le supérieur professionnel, Mr. Jeanmaire Alain).

## Programme maximisé

Si aucune application n'est ouverte, la fenêtre de suivi sera vide.

Si une ou plusieurs applications sont ouvertes, elles s'afficheront, comme le montre la figure ci-dessous. L'affichage montre le nom de l'application, suivi des divers processus ouverts de cette application. Chaque processus est décrit par son numéro de processus, du nom de la fenêtre active et du nom de l'application.

Si plusieurs instances de la même application sont ouvertes elles apparaîtront sous formes de branches du même programme. Les fenêtres-filles n'apparaissent pas. L'arbre est toujours entièrement déployé (pas besoin d'ouvrir les branches).



**Figure 46 : Aperçu du programme maximisé**

Vous pouvez cliquer sur un processus. Dans ce cas, le détail de ce processus s'affichera en bas de la fenêtre (Nom, ID, durée d'utilisation et titre, *ici* : *Google*).

## Conclusion

Au début de ce projet, j'ai pensé qu'aucunes réelles difficultés n'étaient présentes car j'avais déjà expérimenté tous les points qui m'étaient demandé. Mais lorsque je me suis confronté à la gestion des applications et des titres, j'ai vite réalisé que j'allais passer beaucoup de temps à passer ce cap. Et effectivement, j'ai eu de la peine à mettre en place un bon système de gestion, je voulais limiter l'utilisation gourmande des HashTables. Je suis vraiment satisfait du résultat que j'ai obtenu et l'appui de mon supérieur professionnel n'y est pas pour rien.

Au sujet de l'analyse fonctionnelle il y'a une chose que j'ai appris sur moi-même c'est que je préfère avoir une grande liberté lorsque je développe. Je dis ça car j'ai constaté que l'analyse fonctionnelle c'est vite révélé comme un poids plus qu'une aide.

Lorsque j'ai reçu le cahier des charges du projet, je n'avais pas très bien saisi l'intérêt de ces applications cumulées, c'est au fur et à mesure que je découvrais les possibilités de l'application C# que je me suis rendu compte que ses possibilités allaient être des obstacles pour mon développement notamment la partie de gestion des applications et des titres.

En ce qui concerne l'application WEB, je n'ai eu aucune difficultés pour la réaliser car je suis très aise dans ce langage et je me rends compte également que je commence à bien me familiariser avec le langage C#.

Mon objectif principal était de pouvoir pousser au maximum les capacités des applications, je suis un peu déçu de ce côté-là car j'aurai voulu avoir plus de temps pour pouvoir faire mieux notamment en rajoutant une mise à jour automatique du client CIFOM-Spy et également une page PHP de configuration des applications ainsi que des titres autorisés. Mais je suis tout de même très satisfait du résultat final du projet.

Je me suis laisser surprendre au début par l'envergure du projet, j'ai été très optimiste lors de la réalisation de mon planning, suite à la remarque de mon supérieur professionnel, j'ai corrigé l'erreur mais de nouveau à cause de ma mauvaise appréciation de la gestion des applications et des titres, j'ai pris beaucoup de retard sur le planning.

J'aimerais profiter de cette conclusion pour remercier mon supérieur professionnel Mr. Jeanmaire Alain pour m'avoir confié ce projet et d'avoir eu confiance en mes compétences.

Lieu : \_\_\_\_\_

Date : \_\_\_\_\_

Signature : \_\_\_\_\_

## Bibliographie

Pour réaliser ce projet voici les adresses internet qui m'ont été utiles.

Aide PHP sur : <http://www.nexen.net>

Aide C# sur : <http://msdn.microsoft.com>

Hook référence : <http://www.csharpfr.com/code.aspx?ID=20628>

Compte à rebours : <http://www.espacejavascript.com/script.php?id=29>

Adresse de TYPSoft FTP Serveur 1.10 : <http://php.typsoft.com/download.php?id=1>

Classe WindowsDesktop : [http://www.csharpfr.com/codes/CAPTURE-ECRAN-NET\\_18327.aspx](http://www.csharpfr.com/codes/CAPTURE-ECRAN-NET_18327.aspx)

Classe ReqFTP : <http://www.csharpfr.com/archives/archive9.html>

## Annexe

1. Code source de l'application WEB CIFOM-Master
2. Code source de l'application C# CIFOM-Spy
3. Journal de travail
4. Planning