

Πτυχιακή εργασία του φοιτητή Γουναρίδη Δημητρίου-Στυλιανού



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΕΥΚΟΛΗ ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ WINDOWS ΑΠΟ ΑΡΧΑΡΙΟΥΣ ΧΡΗΣΤΕΣ



Του φοιτητή

Γουναρίδη Δημητρίου-Στυλιανού

Αρ. Μητρώου: 133997

Επιβλέπων καθηγητής

Γιακουστίδης Κωνσταντίνος

Θεσσαλονίκη 2017

ΠΡΟΛΟΓΟΣ

Οι υπολογιστές έχουν εισχωρήσει πλέον για τα καλά στην καθημερινότητα των ανθρώπων. Σχεδόν σε κάθε σπίτι υπάρχει τουλάχιστον ένας επιτραπέζιος ή φορητός υπολογιστής που χρησιμοποιεί το δημοφιλές λειτουργικό σύστημα Microsoft Windows. Όμως για τον αρχάριο χρήστη η διαχείριση και συντήρηση ενός τέτοιου συστήματος δεν είναι πάντα εύκολη, με αποτέλεσμα να καταφεύγει είτε σε άτομα με κάποιο παραπάνω τεχνικό υπόβαθρο είτε σε χρήση third-party closed-source εφαρμογών με αμφίβολα αποτελέσματα. Για αυτό το λόγο κρίνεται απαραίτητη και αναγκαία η δημιουργία ενός δωρεάν και open-source εργαλείου που θα διευκολύνει τον αρχάριο (και όχι μόνο) χρήστη να πραγματοποιήσει εύκολα και γρήγορα εργασίες συντήρησης στο Windows σύστημά του αυξάνοντας τις επιδόσεις του, όταν οι διαδικασίες τελειώσουν.

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία θα μελετήσουμε εκτενώς το δημοφιλές λειτουργικό σύστημα Microsoft Windows και τις παραμέτρους που επηρεάζουν την επίδοσή του. Θα αναλύσουμε τα πιο σημαντικά εργαλεία που υπάρχουν ενσωματωμένα στο λειτουργικό σύστημα ενώ παράλληλα θα εξηγήσουμε πως σχεδιάσαμε και αναπτύξαμε ένα εργαλείο, κάνοντας χρήση της γλώσσας C# και του .NET Framework, που επιτρέπει στον αρχάριο χρήστη να εκτελέσει εργασίες συντήρησης και βελτιστοποίησης του συστήματός του με απλό και φιλικό τρόπο μέσω της εφαρμογής μας.

ABSTRACT

In this thesis we will study in detail about the popular Microsoft Windows Operating System as well as its settings that play an important role in its performance. We will also cover some of the internal tools used for administrating the system. Finally we will show and explain how we designed a tool using the C# programming language and .NET Framework to allow the non-expert user to perform maintenance and optimization tasks in a simple and user friendly way.

ΕΥΧΑΡΙΣΤΙΕΣ

Για την πραγματοποίηση της συγκεκριμένης πτυχιακής εργασίας θα ήθελα να ευχαριστήσω τους γονείς μου, που φρόντισαν από όταν ήμουν ακόμα μικρός σε ηλικία να υπάρχει το ερέθισμα των ηλεκτρονικών υπολογιστών μέσα στο σπίτι, τους φίλους μου που πάντα με εμπιστεύονταν για την επίλυση των τεχνικών τους προβλημάτων καθώς και τους καθηγητές της σχολής και ιδιαίτερως τον επιβλέποντα καθηγητή μου Γιακουστίδη Κωνσταντίνο για όλες τις γνώσεις και συμβουλές που μου παρείχαν κατά τη διάρκεια της φοίτησής μου στη σχολή.

Κατάλογος περιεχομένων

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ.....	5
ΕΙΣΑΓΩΓΗ.....	10
ΚΕΦΑΛΑΙΟ 1.....	11
ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ MICROSOFT WINDOWS.....	11
ΕΙΣΑΓΩΓΗ.....	11
1.1 Σύντομη Ιστορική Αναδρομή.....	11
1.2 Η αρχιτεκτονική Windows NT.....	12
1.3 Το Μητρώο των Windows (Windows Registry).....	14
1.4 Η υποδομή Windows Management Instrumentation (WMI).....	18
1.5 Η τεχνολογία των Installers.....	20
1.6 Κοινά προβλήματα των Windows συστημάτων.....	24
ΕΠΙΛΟΓΟΣ.....	31
ΚΕΦΑΛΑΙΟ 2.....	32
ΣΗΜΑΝΤΙΚΑ ΕΡΓΑΛΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ WINDOWS ΣΥΣΤΗΜΑΤΩΝ.....	32
ΕΙΣΑΓΩΓΗ.....	32
2.1 Ο διάλογος Εκτέλεσης (Run Prompt).....	32
2.2 Η Γραμμή Εντολών (Command Prompt).....	33
2.3 Το εργαλείο regedit (Registry Editor).....	33
2.4 Το εργαλείο powercfg (Power Config).....	35
2.5 Το εργαλείο cleanmgr (Disk Cleanup).....	37

2.6 Το εργαλείο services.msc (Services).....	38
2.7 Το εργαλείο msconfig.....	39
ΕΠΙΛΟΓΟΣ.....	42
ΚΕΦΑΛΑΙΟ 3.....	43
ΕΙΣΑΓΩΓΗ ΣΤΟ MICROSOFT .NET FRAMEWORK.....	43
ΕΙΣΑΓΩΓΗ.....	43
3.1 Εισαγωγή στο .NET Framework.....	43
3.2 Σημαντικά συστατικά του .NET Framework.....	44
ΕΠΙΛΟΓΟΣ.....	47
ΚΕΦΑΛΑΙΟ 4.....	48
ADVANCED SYSTEM MANAGER – ΑΝΑΛΥΣΗ ΟΝΤΟΤΗΤΩΝ.....	48
ΕΙΣΑΓΩΓΗ.....	48
Η κλάση MyLogger.....	48
Η κλάση ArgumentParser.....	48
Η κλάση SystemInfo.....	49
Η κλάση CPUPower.....	50
Η κλάση WindowsService.....	50
Η κλάση Service Manager.....	50
Η κλάση Disk Cleanup.....	50
Η κλάση Package.....	50
Η κλάση PackageManager.....	51
Η κλάση RegistryParser.....	53
ΕΠΙΛΟΓΟΣ.....	57
ΚΕΦΑΛΑΙΟ 5.....	58

ΣΥΓΚΡΙΣΗ ΤΟΥ ADVANCED SYSTEM MANAGER ΜΕ ΑΛΛΕΣ ΛΥΣΕΙΣ ΤΟΥ ΕΜΠΟΡΙΟΥ – ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΕΠΕΚΤΑΣΗ.....	58
ΕΙΣΑΓΩΓΗ.....	58
Σύγκριση του Advanced System Manager με άλλα διαθέσιμα optimization εργαλεία.....	58
Προτάσεις για μελλοντική επέκταση της εφαρμογής.....	60
ΕΠΙΛΟΓΟΣ.....	60
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	61
ΑΝΑΦΟΡΕΣ.....	62
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	63

Ευρετήριο πινάκων

Πίνακας 1 :Τύποι Δεδομένων στο μητρώο.....	14
Πίνακας 2: Ονόματα και Περιγραφή Κυψελών Μητρώου.....	15
Πίνακας 3: Κλάσεις της WMI.....	18
Πίνακας 4: Οι εντολές του εργαλείου powercfg.....	37

Ευρετήριο Σχημάτων

Εικόνα 1: Στιγμιότυπο οθόνης από Windows 1.0.....	12
Εικόνα 2: Η καρτέλα Σύστημα.....	24
Εικόνα 3: Δείκτης Εμπειρίας των Windows.....	25
Εικόνα 4: Πληροφορίες Συστήματος.....	26
Εικόνα 5: Το εργαλείο Device Manager.....	27
Εικόνα 6: Έλλειψη Δυνατότητας Μαζικής Απεγκατάστασης.....	28
Εικόνα 7: Ρυθμίσεις Οπτικών Εφέ.....	29
Εικόνα 8: Ρυθμίσεις Ενέργειας των Windows.....	31

Εικόνα 9: Ο διάλογος Εκτέλεσης (Run Prompt).....	32
Εικόνα 10: Η Γραμμή Εντολών (Command Prompt).....	33
Εικόνα 11: Το εργαλείο regedit.....	34
Εικόνα 12: Το γραφικό περιβάλλον ρυθμίσεων ενέργειας των Windows.....	35
Εικόνα 13: Το γραφικό περιβάλλον του εργαλείου Disk Cleanup.....	37
Εικόνα 14: Το γραφικό περιβάλλον του εργαλείου services.....	38
Εικόνα 15: Το εργαλείο msconfig.....	39
Εικόνα 16: Σύγκριση με AdvancedSystemManager.....	40
Εικόνα 17: Η καρτέλα διαχείρισης υπηρεσιών του εργαλείου msconfig.....	41
Εικόνα 18: Χρήση της WMI από τη C# - Παράδειγμα WQL ερωτήματος.....	49
Εικόνα 19: Η καρτέλα SystemInfo.....	65
Εικόνα 20: Η καρτέλα Tools.....	66
Εικόνα 21: Η καρτέλα Startup Manager.....	67
Εικόνα 22: Η καρτέλα Optimize.....	70

ΕΙΣΑΓΩΓΗ

Σκοπός της πτυχιακής εργασίας είναι η δημιουργία ενός δωρεάν και open-source εργαλείου χρησιμοποιώντας τη γλώσσα C# και το .NET Framework (2.0). που επιτρέπει στο χρήστη να πραγματοποιεί εργασίες συντήρησης και βελτίωσης στο Windows σύστημά του.

Στα κεφάλαια που θα ακολουθήσουν, θα πραγματοποιηθεί μια εισαγωγή στο λειτουργικό σύστημα Microsoft Windows, θα παρουσιάσουμε τα πιο κοινά προβλήματα που εμφανίζονται και κάποια σημαντικά εργαλεία διαχείρισης. Στη συνέχεια θα αναλύσουμε τα βασικά συστατικά του .NET Framework και τις πιο σημαντικές κλάσεις του που χρησιμοποιήθηκαν στην υλοποίηση αυτής της εργασίας. Έπειτα θα αναλύσουμε μερικά από τα πιο σημαντικά κομμάτια του κώδικα και πώς συνδυάζοντας τα δύο προηγούμενα κεφάλαια καταλήξαμε στη δημιουργία του εργαλείου της παρούσας πτυχιακής. Θα πραγματοποιήσουμε σύγκριση της εφαρμογής μας με άλλες διαθέσιμες στο εμπόριο και τέλος θα κλείσουμε με έναν σύντομο αλλά επεξηγηματικό οδηγό τής εφαρμογής μας με εικόνες και παραδείγματα που θα διευκολύνουν ακόμα περισσότερο στην χρήση του εργαλείου.

ΚΕΦΑΛΑΙΟ 1

ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ MICROSOFT WINDOWS

ΕΙΣΑΓΩΓΗ

Ο σκοπός του πρώτου κεφαλαίου είναι να γίνει μια εισαγωγή στο δημοφιλές λειτουργικό σύστημα Microsoft Windows. Μετά από μια σύντομη ιστορική αναδρομή θα μελετήσουμε τα βασικά της αρχιτεκτονικής του συστήματος Windows και θα γίνει μια παρουσίαση των κοινών προβλημάτων που αντιμετωπίζουν οι αρχάριοι χρήστες κατά τη χρήση του λειτουργικού. Προαπαιτούμενες γνώσεις για αυτό το κεφάλαιο είναι κυρίως η βασική ορολογία των λειτουργικών συστημάτων. Ακόμα κρίνεται επιθυμητή (αλλά όχι απαραίτητη) η εμπειρία χρήσης ηλεκτρονικού υπολογιστή με Windows στο παρελθόν.

1.1 Σύντομη Ιστορική Αναδρομή



Εικόνα 1: Στιγμιότυπο οθόνης από Windows 1.0

Το λειτουργικό σύστημα Microsoft Windows (ή απλα Windows από εδώ και πέρα) είναι μια οικογένεια λειτουργικών συστημάτων που αναπτύσσεται από την Microsoft. Η πρώτη έκδοση του ήταν το Νοέμβριο του 1985 με την ονομασία Windows 1.0. Ήταν η βελτίωση από το παλιότερο MS-DOS (Microsoft Disk Operating System) που πλέον πρόσφερε στο χρήστη τη δυνατότητα να αλληλεπιδρά με τον υπολογιστή του μέσω ενός GUI (Graphical User Interface). Πρόκειται για ουσιαστικά ένα κέλυφος (shell) που “τρέχει” πάνω σε μια ήδη υπάρχουσα εγκατεστημένη έκδοση του MS-DOS.[1]

Τα Windows αποδεικνύονται ιδιαίτερη επιτυχία για την Microsoft που κατάφερε να εδραιώσει το δικό της γραφικό περιβάλλον/λειτουργικό έναντι του Mac OS της Apple και να αποτελεί μέχρι και σήμερα το δημοφιλέστερο λειτουργικό σύστημα προσωπικών υπολογιστών στον κόσμο.

Ακολουθεί πλήθος νεότερων εκδόσεων με σημαντικούς σταθμούς τα Windows 95(Αύγουστος 1995 – 32/16bit υβριδική αρχιτεκτονική πλέον έναντι των 16bit), τα Windows NT (New Technology) το πρώτο 32bit λειτουργικό σύστημα της Microsoft(οι σημερινές εκδόσεις των Windows συνεχίζουν και θεωρούνται βελτιωμένες εκδοχές της ίδιας αρχιτεκτονικής NT), τα Windows XP (Οκτώβριος 2001), Windows 7 (Ιούλιος 2009) και Windows 10 (Σεπτέμβριος 2014).

1.2 Η αρχιτεκτονική Windows NT

Τα Windows NT (κάθε έκδοση των Windows μετά το 2000 ανήκει στην οικογένεια λειτουργικών NT) είναι η σειρά των λειτουργικών συστημάτων της Microsoft που χρησιμοποιείται μέχρι και σήμερα. Σε αντίθεση με την παλαιότερη αρχιτεκτονική που βασιζόταν στο MS-DOS τώρα έχουμε ένα μοντέρνο λειτουργικό σύστημα διαχωρισμένο σε στρώσεις (layers). Οι δύο σημαντικότερες στρώσεις ονομάζονται “User Mode” και “Kernel Mode” αντίστοιχα. Κάθε mode έχει τα δικά του υποπρογράμματα και υποσυστήματα. Στο user mode όμως τα προγράμματα έχουν περιορισμένη πρόσβαση όσον αφορά το σε ποιους πόρους του συστήματος έχουν πρόσβαση. Αντίθετα οτιδήποτε τρέχει σε kernel mode έχει απεριόριστη πρόσβαση σε όλη τη μνήμη και τις εξωτερικές συσκευές. Επίσης το kernel mode εμποδίζει τις υπηρεσίες και τα προγράμματα που εκτελούνται στο user mode να αποκτήσουν πρόσβαση σε πόρους που δεν έχουν εξουσιοδότηση. Ο πυρήνας(kernel) των Windows NT ανήκει στην κατηγορία των hybrid kernels και περιλαμβάνει τον ίδιο τον πυρήνα, ένα Hardware Abstraction Layer (HAL) και μια σειρά υπηρεσιών που καλούνται για συντομία Executive[1,2].

Όπως ήδη αναφέραμε κάθε τόσο το user mode όσο και το kernel mode περιέχουν υποσυστήματα. Εδώ θα αναφερθούμε στα πιο σημαντικά από αυτά:

Το υποσύστημα Win32 : Αυτό το υποσύστημα αποτελεί μέρος του user mode και επιτρέπει την εκτέλεση 32bit εφαρμογών. Παράλληλα είναι υπεύθυνο για την προσωμείωση των 16bit εφαρμογών στα 32bit συστήματα καθώς και για την διαχείριση παραθύρων (Window Management). Επίσης διαχειρίζεται το input και το προωθεί κατάλληλα στις εφαρμογές.

Στο kernel mode από την άλλη θα αναφερθούμε στα εξής υποσυστήματα:

Executive: Τα Windows Executive Services αποτελούν μια σειρά από χαμηλού επιπέδου (low level) υπηρεσίες που σκοπό έχουν τη διαχείριση του I/O, αντικειμένων, ασφάλειας και διαδικασιών (process management). Μέρος των executive αποτελεί ο Configuration Manager ο οποίος υλοποιεί το μητρώο των Windows (Windows Registry) το οποίο θα απασχολήσει σημαντικό μέρος της εργασίας μας παρακάτω. Άλλα μέρη των executive είναι και ο Memory Manager, Power Manager και ο I/O Manager. Δηλαδή στο kernel mode και μέσα στα executives βλέπουμε να υλοποιούνται μερικές από τις πιο σημαντικές λειτουργίες ενός λειτουργικού συστήματος.

Εκτός των executive στο kernel mode υπάρχει και ο ίδιος ο kernel των Windows, οι drivers του (από τους πολύ υψηλού επιπέδου όπως για παράδειγμα οι οδηγίες συστημάτων αρχείων NTFS, μέχρι τους πολύ χαμηλούς όπως αυτοί των ελεγκτών διαύλων) και τέλος το Hardware Abstraction Layer (HAL), μια “στρώση” ανάμεσα στο hardware του υπολογιστή και το υπόλοιπο λειτουργικό σύστημα. Το HAL έχει ως σκοπό να αποκρύπτει τις διαφορές ανάμεσα στα διάφορα είδη hardware με στόχο την απρόσκοπτη λειτουργία του συστήματος.

1.3 Το Μητρώο των Windows (Windows Registry)

Το Windows Registry (Μητρώο των Windows) είναι μια ιεραρχική βάση δεδομένων που περιέχει δεδομένα τα οποία είναι απαραίτητα για την σωστή λειτουργία του λειτουργικού συστήματος Windows καθώς και δεδομένα για τις εφαρμογές και τις υπηρεσίες (services) που τρέχουν σε αυτό. Τα δεδομένα είναι αποθηκευμένα με μορφή δέντρου. Κάθε κόμβος (node) στο δέντρο ονομάζεται κλειδί (key). Κάθε κλειδί μπορεί να περιέχει και άλλα υποκλειδιά (subkeys) ή και τιμές δεδομένων (values). Είναι δυνατή η ύπαρξη ενός κλειδιού χωρίς καθόλου υποκλειδιά ή δεδομένα. Πολλές φορές μάλιστα τέτοια είδους κλειδιά χρησιμοποιούνται από τις διάφορες εφαρμογές. Στις συγκεκριμένες περιπτώσεις ενδιαφέρει μόνο η ύπαρξη του κλειδιού για να πραγματοποιηθεί μια ενέργεια και όχι κάποια τιμή.

Ένα κλειδί μπορεί να έχει οποιονδήποτε αριθμό τιμών και οι τιμές αυτές μπορούν να είναι αυτές παρακάτω[3]:

Πίνακας 1 :Τύποι Δεδομένων στο μητρώο

Τύπος Δεδομένων	Περιγραφή
REG_BINARY	Ωμά(raw) δυαδικά δεδομένα(binary data). Οι πιο πολλές πληροφορίες για το hardware αποθηκεύονται σε αυτή τη μορφή. Μέσα στον Registry Editor εμφανίζονται σε δεκαεξαδική μορφή.
REG_DWORD	Η αναπαράσταση ενός αριθμού με μήκος 32 bit(4 bytes).
REG_EXPAND_SZ	Ένα String μεταβλητού μήκους. Μπορεί να περιέχει μεταβλητές που χρησιμοποιούνται από προγράμματα ή υπηρεσίες.
REG_MULTI_SZ	Ένα πολλαπλό String,δηλαδή πολλά Strings μαζί χωρισμένα με κάποιο σύμβολο (, . ;)
REG_SZ	Ένα απλό String σταθερού μήκους.
REG_NONE	Δεδομένα χωρίς κάποιον συγκεκριμένο τύπο. Αυτά τα δεδομένα γράφονται στο μητρώο από το σύστημα ή κάποια εφαρμογή και εμφανίζονται στον Registry Editor σαν δυαδικές τιμές σε δεκαεξαδική μορφή.
REG_QWORD	Ένας αριθμός με μήκος 64bit (8 bytes).

Κάθε κλειδί έχει όνομα που αποτελείται από εκτυπώσιμους χαρακτήρες(εκτός από το χαρακτήρα '\ - χωρίς τα ' '). Τα ονόματα των κλειδιών δεν είναι case-sensitive. Τα ονόματα των τιμών και οι ίδιες οι τιμές μπορούν να περιλαμβάνουν και το χαρακτήρα '\.

Κάθε υποκλειδί έχει μοναδικό όνομα σε σχέση με τον πρώτο γονέα του στην ιεραρχία.

Τα ονόματα των κλειδιών επίσης δεν μεταφράζονται αν χρησιμοποιείται μια localized/μεταφρασμένη έκδοση των Windows. Ωστόσο οι τιμές είναι δυνατόν να είναι μεταφρασμένες.

Το Windows Registry αποτελείται από “κυψέλες” (hives). Μια hive είναι ένα σύνολο από κλειδιά, υποκλειδιά και τιμές που επηρεάζουν συγκεκριμένους τομείς του συστήματος. Αντίγραφα αυτών των κυψελών υπάρχουν σε αρχεία τα οποία το λειτουργικό σύστημα φορτώνει κατά την εκκίνηση. Τα αρχεία αυτά βρίσκονται στο φάκελο %SystemRoot%\System32\Config.

Τα ονόματα των κυψελών και οι περιγραφές τους μπορούν να βρεθούν στον παρακάτω πίνακα[4]:

Πίνακας 2: Ονόματα και Περιγραφή Κυψελών Μητρώου

Όνομα Κυψέλης	Περιγραφή
HKEY_CURRENT_USER	Περιλαμβάνει τις ρυθμίσεις για τον χρήστη που είναι συνδεδεμένος στο σύστημα. Κάποιες φορές η κυψέλη αναφέρεται σαν “HKCU”.
HKEY_USERS	Περιέχει όλα τα προφίλ ενεργών χρηστών του συστήματος. Το HKCU αποτελεί στην πραγματικότητα ένα υποκλειδί της κυψέλης HKEY_USERS (“HKU”).
HKEY_LOCAL_MACHINE	Περιέχει πληροφορίες για το σύστημα (άρα για όλους τους χρήστες). Η HKEY_LOCAL_MACHINE αναφέρεται και ως “HKLM”.

Όνομα Κυψέλης	Περιγραφή
HKEY_CLASSES_ROOT	Στην πραγματικότητα αποτελεί ουσιαστικά ένα υποκλειδί του HKEY_LOCAL_MACHINE\Software. Εδώ αποθηκεύονται ρυθμίσεις κυρίως για την συσχέτιση των προγραμμάτων και το ποιες επεκτάσεις μπορούν να ανοίγουν. Κάποιες φορές αναφέρεται και σαν "HKCR".
HKEY_CURRENT_CONFIG	Περιέχει πληροφορίες για το hardware προφιλ του υπολογιστή που χρησιμοποιείται κατά την εκκίνηση του συστήματος.

Στις 64-bit εκδόσεις των Windows (XP και έπειτα) το Registry χωρίζεται σε δύο μέρη με 32 και 64bit αντίστοιχα. Πολλά από τα κλειδιά του 32bit τμήματος μοιράζονται ίδια ονόματα με αυτού του 64bit τμήματος και το αντίστροφο. Σε ένα λειτουργικό σύστημα 64bit τα 32bit κλειδιά μπορούν να εντοπιστούν στη διαδρομή: **HKEY_LOCAL_MACHINE\Software\WOW6432Node**

Το WoW64 (**W**indows 32-bit **on** **W**indows 64-bit) αποτελεί ένα υποσύστημα των Windows που επιτρέπει την εκτέλεση 32bit εφαρμογών στις 64bit εκδόσεις του λειτουργικού.

Το WoW64 αναλαμβάνει να λύσει τα προβλήματα συμβατότητας των 32bit εφαρμογών που τρέχουν σε ένα 64bit λειτουργικό Windows. Όσον αφορά το Registry το κλειδί HKEY_LOCAL_MACHINE\Software\WOW6432Node πρόκειται ουσιαστικά για μια 32bit only αναπαράσταση/ανακατεύθυνση του HKEY_LOCAL_MACHINE\Software που θα υπήρχε σε ένα 32bit σύστημα. Οι εφαρμογές 32bit που χρειάζονται αυτό το κομμάτι του Registry γράφουν και διαβάζουν στο HKEY_LOCAL_MACHINE\Software\WOW6432Node "νομίζοντας" πως πρόκειται για το HKEY_LOCAL_MACHINE\Software που περίμεναν να βρουν. Αυτή η 'παραπλάνηση' των εφαρμογών γίνεται από το WoW64. Έτσι αποφεύγεται στα 64bit συστήματα η χρήση του κλειδιού HKEY_LOCAL_MACHINE\Software από τις 32bit εφαρμογές και κατ'επέκταση εκεί βρίσκονται μόνο 64bit εφαρμογές και επιτυγχάνεται διαχωρισμός των κλειδιών-υποσυστημάτων. Για να το πετύχει αυτό το WoW64 χρησιμοποιεί έναν "Registry

Reflector” που ουσιαστικά αντανακλά κλειδιά και τιμές ανάμεσα στις 32 και 64bit όψεις του Registry.

Το WoW64 πραγματοποιεί αντίστοιχες ενέργειες και με το φάκελο %SystemRoot%\system32 όπου πάλι ‘παραπλανεί’ και παρουσιάζει στις 32bit εφαρμογές τον φάκελο %SystemRoot%\SysWow64 όπου εκεί υπάρχουν οι 32bit εκδόσεις πολλών .dll και εργαλείων των Windows(πχ η 32bit έκδοση του cmd.exe).

Σε όλες τις εκδόσεις των Windows υπάρχουν στο Registry κλειδιά με πληροφορίες για το εγκατεστημένο λογισμικό στο σύστημα, συμπεριλαμβανομένων και των Windows Updates. Οι πληροφορίες αυτές συμβολίζονται στο Registry σαν τιμές (values) διαφόρων τύπων. Κάθε λογισμικό αποτελεί ένα κλειδί και οι τιμές που περιλαμβάνει είναι βασικά στοιχεία όπως το όνομα και το μέγεθος που καταλαμβάνει στο δίσκο μέχρι περισσότερο σύνθετα στοιχεία όπως ένα Uninstall String(μια εντολή που αν εκτελεστεί θα αφαιρεθεί το συγκεκριμένο λογισμικό). [9,10]

Παρακάτω στο κεφάλαιο 4 θα δούμε πώς το πρόγραμμά μας εξετάζει τα στοιχεία του εγκατεστημένου λογισμικού στο σύστημα για να σχηματίσει τη λίστα των διαθέσιμων προς αφαίρεση προγραμμάτων.

1.4 Η υποδομή Windows Management Instrumentation (WMI)

Η Windows Management Instrumentation (WMI) αποτελεί μια ενσωματωμένη υποδομή των λειτουργικών συστημάτων Microsoft Windows για διαχείριση δεδομένων και εκτέλεση εργασιών σχετικών με το λειτουργικό σύστημα. [5]

Η WMI αποτελεί ουσιαστικά την υλοποίηση της Microsoft για τα πρότυπα WBEM και CIM (Web-Based Enterprise Management και Common Information Model αντίστοιχα), δύο πρότυπα που χρησιμοποιούνται ευρέως στις επιχειρήσεις για διαχείριση πόρων. Η WMI χρησιμοποιεί το CIM για την αναπαράσταση συστημάτων, συσκευών, δικτύων και άλλων τμημάτων διοίκησης.

Οι κλήσεις στις μεθόδους της WMI πραγματοποιούνται με χρήση WQL ερωτημάτων. Η WQL (Windows Management Instrumentation Query Language) είναι μια γλώσσα που αναπτύχθηκε από τη Microsoft για χρήση στη WMI και αποτελεί υποσύνολο (subset) του ANSI προτύπου της SQL . Η ομοιότητα με την SQL θα γίνει εμφανής στο παράδειγμα παρακάτω.

Η WMI προσφέρει τη δυνατότητα στους χρήστες/διαχειριστές για χρήση των δεδομένων της ή και εκτέλεση εργασιών μέσω scripts ή γλωσσών προγραμματισμού όπως με το Windows PowerShell και τη C#. Η WMI φέρνει και

ένα command line interface το WMI (Windows Management Instrumentation Command-Line) αλλά εμείς θα τη χρησιμοποιήσουμε στην εφαρμογή μας μέσω της C# και του .NET.

Η WMI παρέχει εκτός των άλλων πληροφορίες για το hardware που υπάρχει διαθέσιμο στο σύστημα μας μέσω κλάσεων (classes). Κάθε κλάση αποτελεί μια αναπαράσταση μιας συσκευής, μιας λειτουργίας, μιας οντότητας. Κάθε κλάση έχει συγκεκριμένα πεδία/ιδιότητες συγκεκριμένου τύπου (int, string κτλ) ακριβώς όπως οι κλάσεις σε μια γλώσσα προγραμματισμού.

Οι κλάσεις που θα χρησιμοποιήσουμε στο πρόγραμμα μας και η λειτουργία τους καθώς και τα πεδία που θα χρησιμοποιήσουμε φαίνονται στον παρακάτω πίνακα:

Πίνακας 3: Κλάσεις της WMI

Όνομα Κλάσης	Περιγραφή	Πεδία
Win32_OperatingSystem	Αναπαριστά ένα Windows-based λειτουργικό σύστημα που βρίσκεται εγκατεστημένο στον υπολογιστή μας.	OSArchitecture για την εύρεση της αρχιτεκτονικής του λειτουργικού (32 ή 64bit)
Win32_BaseBoard	Αναπαριστά τη συσκευή της μητρικής κάρτας motherboard που βρίσκεται σε έναν υπολογιστή.	Product για την εξαγωγή του μοντέλου της μητρικής που υπάρχει εγκατεστημένη στο σύστημά.
Win32_Processor	Αναπαριστά μια συσκευή που μπορεί να ερμηνεύει αλληλουχίες οδηγιών(δηλαδή έναν επεξεργαστή).	Name για την εξαγωγή του ονόματος/μοντέλου του επεξεργαστή
Win32_ComputerSystem	Αναπαριστά ένα ολοκληρωμένο υπολογιστικό σύστημα.	TotalPhysicalMemory για την εξαγωγή του μεγέθους της μνήμης RAM που είναι εγκατεστημένη στο σύστημα εκφρασμένη σε Kilobytes.

Όνομα Κλάσης	Περιγραφή	Πεδία
Win32_VideoController	Αναπαριστά έναν ελεγκτή Video(δηλαδή μία κάρτα γραφικών).	Name και AdapterRAM για εύρεση του ονόματος της κάρτας γραφικών και της ποσότητας της αυτόνομης μνήμης γραφικών εκφρασμένη σε Kilobytes.
Win32_DiskDrive	Αναπαριστά μια συσκευή σκληρού δίσκου.	Model και Size για εύρεση του ονόματος του σκληρού δίσκου που βρίσκεται εγκατεστημένος στο σύστημα μας και του συνολικού μεγέθους του εκφρασμένο σε Kilobytes.

Ωστόσο αν και η WMI μπορεί να προσφέρει πολλές ακόμα πληροφορίες όπως για τα ενεργά services ή τα προγράμματα που είναι εγκατεστημένα στο σύστημα μας, δεν είναι πανάκεια. Αυτό γιατί έρχεται με μεγάλο κόστος στην απόδοση του προγράμματος μας. Για αυτό και στο πρόγραμμα μας χρησιμοποιείται μόνο για την απόκτηση βασικών πληροφοριών για το hardware του συστήματος μας και για την απόκτηση των υπόλοιπων πληροφοριών (services, startup items, προγράμματα) χρησιμοποιούμε το μητρώο των Windows (Windows Registry) για να έχουμε πρόσβαση σε αυτές.

1.5 Η τεχνολογία των Installers

Ως **Installer** ορίζεται ένα πρόγραμμα εγκατάστασης (ή αλλιώς setup program) είναι ένα πρόγραμμα που οδηγεί τον χρήστη στη διαδικασία εγκατάστασης ενός άλλου προγράμματος που επιθυμεί να έχει στο σύστημα του συνήθως μέσω γραφικών διαλόγων Next/Next/Next.

Υπάρχουν πολλοί και διαφορετικοί installers για το λειτουργικό σύστημα Microsoft Windows. Εμείς θα αναφέρουμε και θα αναλύσουμε τους πιο γνωστούς και θα δείξουμε πώς πραγματοποιεί την αυτόματη (απ)εγκατάσταση των προγραμμάτων ο AdvancedSystemManager. Οι πιο ευρέως διαδεδομένοι installers είναι οι παρακάτω:

- Windows Installer
- InstallShield
- Nullsoft Scriptable Install System (NSIS)
- Inno Setup
- InstallBuilder

Καθένας από αυτούς τους installers δέχεται διαφορετικές παραμέτρους και αλληλεπιδρά διαφορετικά με το σύστημα. Επίσης διατηρεί αποθηκευμένο σε διαφορετική μορφή το πρόγραμμα προς εγκατάσταση.

O Windows Installer

Ο Windows Installer αποτελεί ένα software framework και API της Microsoft για την εγκατάσταση, συντήρηση και αφαίρεση λογισμικού από ένα Windows σύστημα. Η εγκατάσταση είναι δυνατή τόσο με τον κλασικό τρόπο με διαλόγους (Next/Next/Next) όσο και από περιβάλλον κονσόλας με χρήση κατάλληλων παραμέτρων. Η Microsoft ενθαρρύνει τη χρήση του Windows Installer έτσι ώστε η εσωτερική βάση των Windows με τον κατάλογο εγκατεστημένων προγραμμάτων να παραμένει συνεπής ως προς την ακεραιότητα των δεδομένων της. Ο Windows Installer πραγματοποιεί transactional ενέργειες στο σύστημα, δηλαδή αν κάτι αποτύχει κατά τη διαδικασία εκτέλεσης του καμία αλλαγή δεν μένει ημιτελής στο σύστημα αλλά επαναφέρεται στην προηγούμενη της κατάσταση. Τα αρχεία εγκατάστασης του Windows Installer έχουν επεκτάσεις .msi και .msp .

Installshiled

Ο InstallShield είναι ένας από τους πιο δημοφιλείς installers και αποτελεί λογισμικό κλειστού κώδικα που αναπτύσσεται από την εταιρεία Flexera Software.

Ο InstallShield έχει τη δυνατότητα να δημιουργήσει και .exe αλλά και .msi εκτελέσιμα αρχεία εγκατάστασης. Προσφέρει κ αυτός ένα κλασικό GUI με Next/Next/Next διαλόγους όπως και παραμέτρους κονσόλας.

Nullsoft Scriptable Install System

Το Nullsoft Scriptable Install System(για συντομία NSIS) είναι ένα σύστημα από scripts από την εταιρεία Nullsoft (δημιουργό του δημοφιλούς Media Player Winamp). Αποτελεί ελεύθερο λογισμικό και έχει γνωρίσει ευρεία αποδοχή από μεγάλες εταιρείες του χώρου. Παράγει αρχεία .exe μόνο αλλά προσφέρει τόσο GUI όσο και παραμέτρους για command line. Τα scripts του και η δυνατότητα για

επέκταση του μέσω plugins τον καθιστούν ιδανικό για πολύπλοκες περιπτώσεις εγκατάστασης λογισμικού. Παράγει αρχεία με επέκταση .exe .

Inno Setup

Το Inno Setup είναι και αυτό ένα σύστημα από scripts που παράγει .exe αρχεία εγκατάστασης. Είναι γραμμένο σε γλώσσα Delphi και είναι ελεύθερο λογισμικό που χρησιμοποιείται από αρκετά γνωστά προγράμματα ανοιχτού και ελεύθερου λογισμικού όπως το GIMP.

InstallBuilder

Κλειστού κώδικα cross-platform δημιουργός αρχείων εγκατάστασης. Δεν χρησιμοποιείται από πολλά προγράμματα και παρουσιάζει μια μικρή ανομοιογένεια όσον αφορά τις παραμέτρους που δέχεται στο περιβάλλον κονσόλας σε σχέση με τους άλλους installers. Θα δούμε περισσότερα για αυτές τις διαφορές παρακάτω. Το γραφικό του περιβάλλον ωστόσο αποτελείται από παραδοσιακούς Next/Next/Next διαλόγους. Παράγει αρχεία με επέκταση .exe .

Unattended Installation

Ως “Unattended Installation” θα μπορούσαμε να ορίσουμε τη διαδικασία μιας εγκατάστασης που ο χρήστης δεν χρειάζεται να αλληλεπιδράσει καθόλου (ή ελάχιστα) με το πρόγραμμα εγκατάστασης. Ο κλασικός τρόπος εγκατάστασης μέσω διαλόγων Next/Next/Next δεν αποτελεί τρόπο Unattended Installation. Αντίθετα αν μια εγκατάσταση πραγματοποιηθεί μέσω ενός απλού διπλού κλικ ή αυτόματα και μαζικά για πολλά προγράμματα μέσω ενός script αυτοματοποίησης τότε θεωρείται Unattended Installation. Σε μια Unattended εγκατάσταση είναι δυνατόν να εμφανίζονται διάλογοι και μπάρες προόδου. Οι απαντήσεις όμως που δίνονται στον installer είναι προκαθορισμένες.

Silent Installation

Ως Silent Installation ορίζεται η διαδικασία μιας εγκατάστασης που συμβαίνουν στο παρασκήνιο, κρυφά από το χρήστη. Αυτό σημαίνει πως εμφανίζονται καθόλου ή ελάχιστοι διάλογοι στο χρήστη. Παρ’όλα αυτά μπορεί να χρειαστεί μια επιβεβαίωση από το χρήστη ακόμα και σε Silent Installation από κάποιους installers.

Συχνά οι όροι Unattended και Silent Installation χρησιμοποιούνται με το ίδιο νόημα. Ωστόσο υπάρχει μια μικρή διαφορά. Μια Unattended εγκατάσταση μπορεί να είναι ορατή στο χρήστη αλλά να μην χρειάζεται κάποια αλληλεπίδραση από

αυτόν. Αντίθετα μια “σιωπηλή” εγκατάσταση μπορεί να χρειαστεί αλληλεπίδραση από το χρήστη σε ορισμένες περιπτώσεις.

Ο AdvancedSystemManger κατά τη διάρκεια της εγκατάστασης/απεγκατάστασης πολλαπλών προγραμμάτων προσπαθεί για ένα συνδυασμό απόκρυψης από το χρήστη τυχόν διαλόγων αλλά και αυτοματοποιημένης εγκατάστασης όπου αυτό καθίσταται δυνατό.

Κάθε τύπος installer δέχεται και διαφορετικές παραμέτρους.

Ο **Windows Installer** για να πραγματοποιήσει μια σιωπηλή εγκατάσταση χρειάζεται να εκτελεστεί από την κονσόλα με την παράμετρο **/qn** .

Παράδειγμα: MsiExec.exe /I /qn Path_Of_MSI_Package.msi

Οι υπόλοιποι installers λίγο πολύ πραγματοποιούν σιωπηλή εγκατάσταση με μία από τις παρακάτω παραμέτρους **/S**, **/Silent**, **/SILENT** . Οι παραπάνω installers με εξαίρεση τον InstallBuilder δεν εμφανίζουν κάποιο μήνυμα λάθους αν δεχθούν άγνωστες παραμέτρους αλλά συνεχίζουν κανονικά την εκτέλεσή τους μόνο με τις γνωστές παραμέτρους. Έτσι είναι εύκολο “χτίζοντας” μέσα στον κώδικα του AdvancedSystemManager μια συμβολοσειρά με όλες τις παραμέτρους που γίνονται αποδεκτές από τους εν λόγω installers να έχουμε μια κοινή εντολή εγκατάστασης για όλους.

Στην περίπτωση του InstallBuilder απλά παρακολουθούμε την έξοδο του προγράμματος για ένα συγκεκριμένο μήνυμα λάθους και τότε επανεκκινούμε τον installer με τις σωστές για τον InstallBuilder παραμέτρους(--mode unattended) .

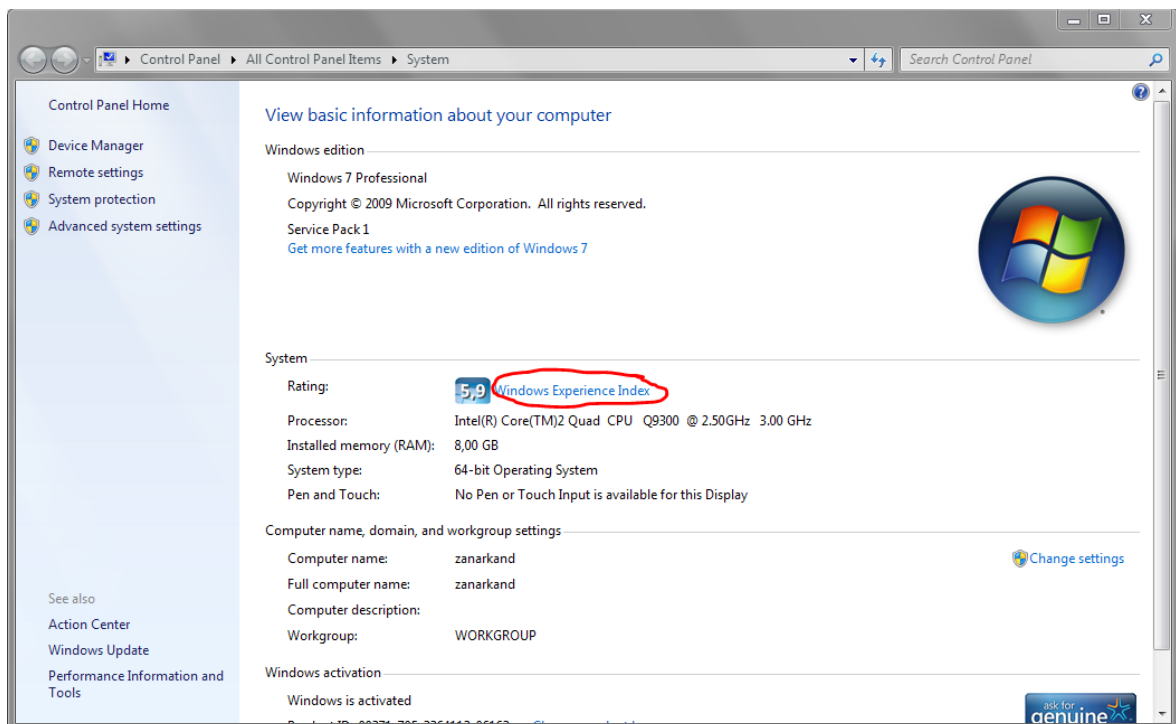
Για τον Windows Installer χρησιμοποιούμε ξεχωριστή εντολή σιωπηλής εγκατάστασης χωρίς να υπάρχει περιθώριο σύγχυσης καθώς α) ο Windows Installer δέχεται πάντα σε όλες τις εκδόσεις του τις ίδιες ακριβώς παραμέτρους και β) τα αρχεία του έχουν επέκταση .msi και άρα το πρόγραμμα μπορεί να τα διακρίνει εύκολα και να τα εκτελέσει κατάλληλα. Μια άλλη προσέγγιση στο θέμα θα ήταν η μελέτη των headers των αρχείων εγκατάστασης πρώτου εκτελεστούν. Αυτό όμως θα αύξανε και την πολυπλοκότητα του κώδικα αλλά και μόνο τα αρχεία του Inno Setup περιέχουν ένα συγκεκριμένο header που μπορεί να χρησιμοποιηθεί για αυτό το σκοπό. Οι υπόλοιποι installers χρησιμοποιούν αρχεία που είναι κοινά Win32 εκτελέσιμα χωρίς κάποια αναφορά ότι πρόκειται για αρχεία εγκατάστασης.

1.6 Κοινά προβλήματα των Windows συστημάτων

Τα πιο κοινά προβλήματα που καλείται να αντιμετωπίσει ένας χρήστης σε ένα Windows σύστημα, αποτελούν λίγο πολύ σχεδιαστικά λάθη της φιλοσοφίας του ίδιου του λειτουργικού συστήματος των Windows. Ας δούμε όμως παρακάτω μερικά από αυτά:

Δυσκολία Εύρεσης Πληροφοριών Συστήματος

Στα Windows υπάρχει έντονη δυσκολία εντοπισμού των πληροφοριών του συστήματος σε πολλά διαφορετικά σημεία (άγνωστα συχνά στον αρχάριο χρήστη) με αποτέλεσμα την άγνοια του χρήστη για τις βασικές hardware πληροφορίες του συστήματός του που μπορεί να κριθούν απαραίτητες σε μια απλή διαδικασία επίλυσης κάποιου τεχνικού προβλήματος. Ενδεικτικά αντί να υπάρχουν όλες οι hardware πληροφορίες σε ένα μόνο σημείο, μπορούν να βρεθούν σε διαφορετικά σημεία και όχι ολοκληρωμένες. Πιο συγκεκριμένα ο χρήστης πρέπει να μεταβεί στην καρτέλα Σύστημα (Πίνακας Ελέγχου->Σύστημα) όπου εκεί βρίσκονται πληροφορίες μόνο για τον επεξεργαστή, την ποσότητα της RAM που βρίσκεται εγκατεστημένη στο σύστημα και την έκδοση του λειτουργικού συστήματος που εκτελείται.



Εικόνα 2: Η καρτέλα Σύστημα

Αν το σύστημα χρησιμοποιεί έκδοση Windows7 ή νεότερη στην καρτέλα σύστημα ο χρήστης μπορεί να επιλέξει τον “Δείκτη Εμπειρίας των Windows” για να μεταβεί σε μια νέα καρτέλα και από εκεί να πατήσει το κουμπί “εκτύπωση πληροφοριών συστήματος” για να αποκτήσει πρόσβαση σε περισσότερες πληροφορίες. Και κει ακόμα όμως λείπουν πληροφορίες για το μοντέλο και τη συνολική χωρητικότητα του σκληρού δίσκου του συστήματος και για το μοντέλο της μητρικής κάρτας ενώ το περιβάλλον δεν είναι και το πιο προσιτό για κατανόηση από έναν αρχάριο χρήστη.

Component	What is rated	Subscore	Base score
Processor:	Calculations per second	7,3	5,9 Determined by lowest subscore
Memory (RAM):	Memory operations per second	7,3	
Graphics:	Desktop performance for Windows Aero	7,8	
Gaming graphics:	3D business and gaming graphics performance	7,8	
Primary hard disk:	Disk data transfer rate	5,9	

What do these numbers mean?
Tips for improving your computer's performance.
Learn more about scores and software online

Your scores are current
Last update: 21/6/2017 4:00:49 μμ

Re-run the assessment

Εικόνα 3: Δείκτης Εμπειρίας των Windows

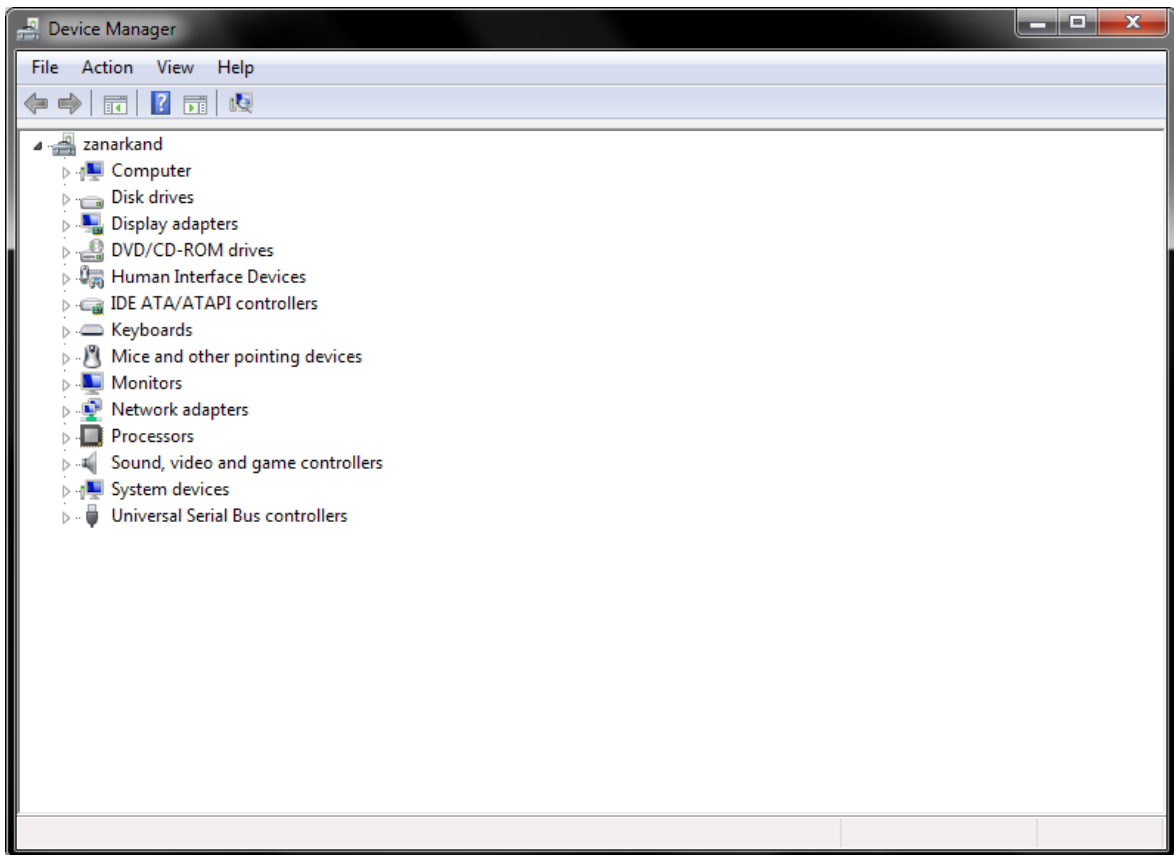
The screenshot shows the 'Performance Information and Tools' window in Windows 7. It displays system performance metrics and a base score of 5.9. The window is titled 'Performance Information and Tools' and has a 'Print this page' button. The main content is divided into sections: 'More details about my computer', 'System', 'Storage', 'Graphics', and 'Network'. The 'More details about my computer' section contains a table with the following data:

Component	Details	Subscore	Base score
Processor	Intel(R) Core(TM)2 Quad CPU Q9300 @ 2.50GHz	7,3	5,9 Determined by lowest subscore
Memory (RAM)	8,00 GB	7,3	
Graphics	AMD Radeon HD 6800 Series	7,8	
Gaming graphics	4863 MB Total available graphics memory	7,8	
Primary hard disk	29GB Free (121GB Total)	5,9	

Below the table, it indicates 'Windows 7 Professional'. The 'System' section lists: Manufacturer (To Be Filled By O.E.M.), Model (To Be Filled By O.E.M.), Total amount of system memory (8,00 GB RAM), System type (64-bit operating system), and Number of processor cores (4). The 'Storage' section lists: Total size of hard disk(s) (401 GB), Disk partition (C:) (29 GB Free (121 GB Total)), Disk partition (D:) (7 GB Free (280 GB Total)), Media drive (X:) (CD/DVD), and Media drive (Y:) (CD/DVD). The 'Graphics' section lists: Display adapter type (AMD Radeon HD 6800 Series), Total available graphics memory (4863 MB), Dedicated graphics memory (1024 MB), Dedicated system memory (0 MB), Shared system memory (3839 MB), Display adapter driver version (15.200.1062.1004), Primary monitor resolution (1680x1050), and DirectX version (DirectX 10). The 'Network' section is partially visible at the bottom.

Εικόνα 4: Πληροφορίες Συστήματος

Επίσης στην περίπτωση των Windows XP & Windows 10 οι παραπάνω αναφερόμενες καρτέλες/εργαλεία δεν υπάρχουν στο σύστημα και άρα ο χρήστης πρέπει να καταλήξει σε κάποιο άλλο εργαλείο των Windows(πχ DeviceManager) για την εύρεση των hardware πληροφοριών του συστήματός του.



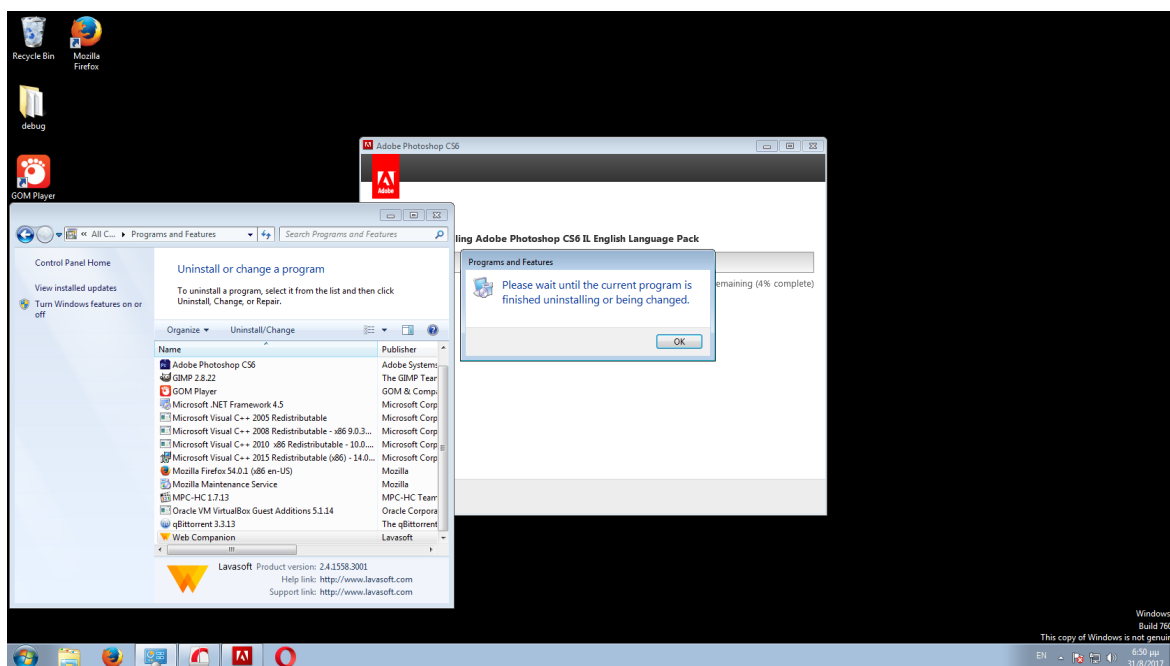
Εικόνα 5: Το εργαλείο Device Manager

Έλλειψη ενός Package Manager

Μια άλλη σοβαρή αδυναμία των Windows είναι η απουσία ενός package manager και ενός software center, στοιχεία που εμφανίζονται στις περισσότερες Linux διανομές ή και στα έξυπνα κινητά τηλέφωνα (smartphones). Στα Windows δεν υπάρχει ένα ενοποιημένο, ασφαλές σημείο που ο χρήστης έχει άμεσα διαθέσιμα πλήθος προγραμμάτων, ελεγμένα και εύκολα προς εγκατάσταση με ένα μόνο κλικ. Αυτό δημιουργεί προβληματισμούς σχετικά με την ασφάλεια ενός συστήματος όταν ο χρήστης μπορεί αυθαίρετα να εγκαθιστά λογισμικό από αμφιβόλου ποιότητας πηγές. Το Windows Store που εμφανίστηκε μαζί με τα Windows 8 και έπειτα δεν λύνει αυτό το πρόβλημα και δεν αποτελεί μια καλή υλοποίηση ενός software center καθώς δεν υπάρχουν ιδιαίτερα δημοφιλή πακέτα (για παράδειγμα Mozilla Firefox, Google Chrome, VLC Media Player, WinRAR) που εγκαθιστούν σχεδόν όλοι οι χρήστες στον υπολογιστή τους.

Επίσης η διεπιφάνεια χρήστη “Προγράμματα και Δυνατότητες” δεν προσφέρει στο χρήστη τη δυνατότητα πολλαπλής επιλογής προγραμμάτων προς αφαίρεση γεγονός που τον υποχρεώνει να απεγκαθιστά ανεπιθύμητα προγράμματα ένα κάθε

φορά. Αυτό δυσχεραίνει την εμπειρία χρήσης ειδικά σε περιπτώσεις συντήρησης ενός συστήματος που το πλήθος των υποψηφίων προγραμμάτων προς αφαίρεση μπορεί να είναι πολύ μεγάλο. Ακόμα δεν δίνεται η δυνατότητα “σιωπηλής απεγκατάστασης” (silent uninstall) , δηλαδή ο χρήστης πρέπει για κάθε πρόγραμμα που θέλει να αφαιρέσει από το σύστημά του να περάσει από μια διαδικασία αποτελούμενη από Next/Next/Next διαλόγους.



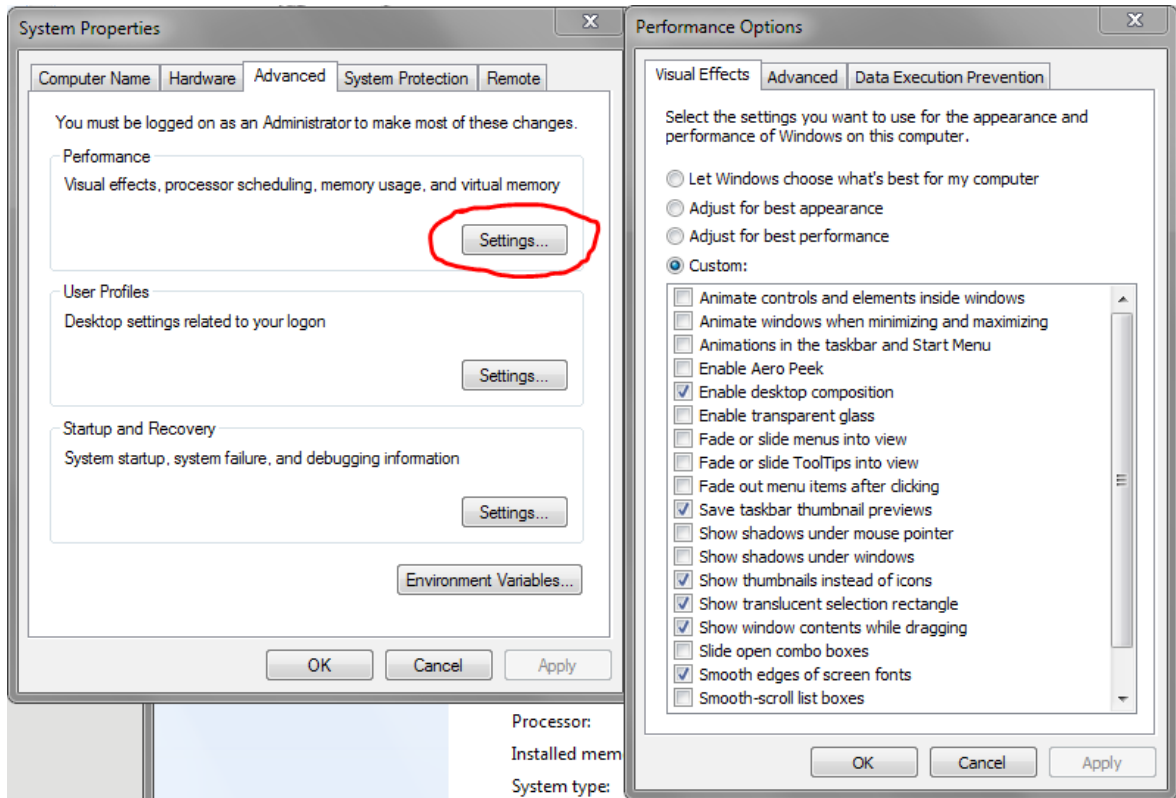
Εικόνα 6: Έλλειψη Δυνατότητας Μαζικής Απεγκατάστασης

Τα οπτικά εφέ των Windows

Ένα ακόμα μειονέκτημα των Windows είναι η απόκρυψη των σχετικών με τις επιδόσεις ρυθμίσεων των οπτικών εφέ.

Από τα Windows XP και έπειτα ο χρήστης πηγαίνοντας στην καρτέλα “Σύστημα” , “Προχωρημένες Ρυθμίσεις Συστήματος” , καρτέλα “Επιδόσεις” μπορεί να διακρίνει πληθώρα ρυθμίσεων για τα οπτικά εφέ που είναι ενεργοποιημένα στο σύστημά του. Τα εφέ αυτά ωστόσο αν και δεν προσδίδουν ιδιαίτερα ορατές αλλαγές στο οπτικό περιβάλλον που χειρίζεται ο χρήστης, εντούτοις μπορούν να προκαλέσουν μικροκολλήματα είτε λόγω αδύναμου hardware είτε εξαιτίας μη υποστήριξης του υλικού του υπολογιστή (για παράδειγμα nvidia drivers – dual monitor setup) .

Τα περισσότερα από τα εφέ απλά προσθέτουν σκιές γύρω από τα παράθυρα και το ποντίκι καθώς και μικρές κινήσεις (animations) στις ενέργειες που πραγματοποιούνται στα παράθυρα. Ένα μη έμπειρο όμως μάτι δεν θα αντιληφθεί εύκολα την απουσία των εν λόγω οπτικών εφέ. Αντίθετα η απόκριση του συστήματος θα ανεβεί σε αντιληπτό βαθμό άμεσα με την κατάργηση αυτών των εφέ.



Εικόνα 7: Ρυθμίσεις Οπτικών Εφέ

Η διαχείριση ενέργειας των Windows - CPU Frequency Scaling

Αν και το επόμενο πρόβλημα δεν αποτελεί τόσο πρόβλημα των Windows αλλά περισσότερο της σύγχρονης τεχνολογίας επεξεργαστών, αποφασίσαμε να το αναφέρουμε ως πρόβλημα σε αυτό το τμήμα και πάλι για λόγους απόκρυψης των ρυθμίσεων.

Οι περισσότεροι σύγχρονοι επεξεργαστές έχουν τη δυνατότητα δυναμικής προσαρμογής του ρολογιού τους (clock rate) για σκοπούς εξοικονόμησης ενέργειας. Η συχνότητα ενός επεξεργαστή σε Ghz δίνεται από τη σχέση:
(bus rate) * multiplier = clock rate.

Ο δίαυλος(bus) γνωρίζει διάφορες υλοποιήσεις ανάλογα με τον κατασκευαστή του επεξεργαστή.

Συνηθισμένοι δίαυλοι σήμερα για x86 επεξεργαστές είναι οι DMI,QPI(Intel) και HyperTransport (AMD) που αντικατέστησαν τους παλαιότερους FSB και EV6 αντίστοιχα.

Ο επεξεργαστής για να είναι "σταθερός" και να μην συμβαίνουν σφάλματα εκτέλεσης χρειάζεται και την σωστή τάση ρεύματος. Ένα υψηλότερο σε συχνότητα ρολόι χρειάζεται μεγαλύτερη τάση απ'ότι ένα με χαμηλότερη. Κατ'επέκταση ένα

χαμηλότερο ρολόι προσφέρει μειωμένες επιδόσεις στην μια πλευρά αλλά και χαμηλότερη κατανάλωση ενέργειας και παραγωγή θερμότητας από την άλλη.

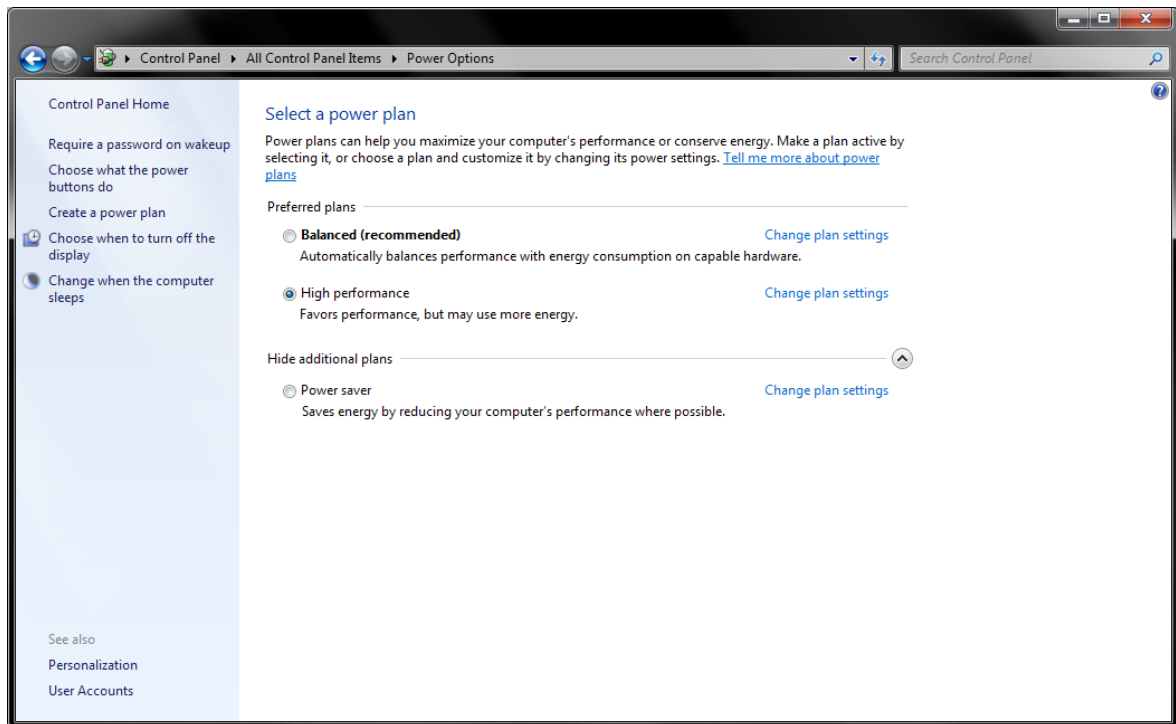
Επειδή ο ρυθμός μετάδοσης του διαύλου δεν μπορεί να τροποποιηθεί από το λειτουργικό σύστημα αυτό που αλλάζει τελικά για να μειωθεί δυναμικά η ταχύτητα ενός επεξεργαστή είναι ο πολλαπλασιαστής (multiplier). Μέσω του ACPI interface το λειτουργικό σύστημα προσαρμόζει 'on the fly' τον πολλαπλασιαστή του επεξεργαστή και την τάση που χρειάζεται οδηγώντας σε εξοικονόμηση ενέργειας και μειωμένη παραγωγή θερμότητας.

Η απόφαση για το πότε και για πόσο θα μειωθεί η συχνότητα ρολογιού του επεξεργαστή διαφέρει από λειτουργικό σε λειτουργικό σύστημα. Στα λειτουργικά Linux η απόφαση αυτή γίνεται μέσω ενός driver(kernel module) του `cpufreq` ο οποίος ανάλογα με τον `cpu frequency governor` που είναι φορτωμένος προσαρμόζει τη συχνότητα. Ο `governor` είναι ένας αλγόριθμος που ελέγχει τον τωρινό φόρτο της CPU και καταγράφοντας παράλληλα στατιστικά φόρτου προσαρμόζεται ανάλογα για να αυξομειώσει τη συχνότητα.

Αντίστοιχα στα Windows τηρείται μια παρόμοια διαδικασία αλλά μέσω του Windows Power Management. Στο λειτουργικό σύστημα των Windows η CPU ορίζεται να λειτουργεί ανάμεσα σε ένα ελάχιστο ποσοστό της μέγιστης συχνότητας και ένα μεγαλύτερο πιο κοντά στη μέγιστη. Για παράδειγμα ένας επεξεργαστής με μέγιστη συχνότητα 3.0GHz αν έχει οριστεί σαν `minimum frequency policy` ένα ποσοστό 25% τότε όταν ο επεξεργαστής είναι `idle`(σε αδράνεια) θα πέφτει στα $3 \cdot 25 / 100 = 0.75$ Ghz ενώ αντίστοιχα στο `maximum frequency policy` αν έχει οριστεί 70% θα αποκτά μια μέγιστη συχνότητα $3 \cdot 70 / 100 = 2.1$ Ghz. Πρακτικά, έχοντας για `minimum frequency scaling` το 100% και φυσικά και μέγιστο το 100% ο επεξεργαστής θα βρίσκεται συνεχώς στην μέγιστη συχνότητα λειτουργίας του (καταναλώνοντας βέβαια περισσότερη ενέργεια και παράγοντας περισσότερη θερμότητα) προσφέροντας μεγαλύτερες επιδόσεις και άμεση απόκριση καθώς δεν σπαταλάται χρόνος για να αυξομειωθεί η συχνότητα της CPU ώστε να καλύπτει το τωρινό φόρτο.

Στα Windows η ρύθμιση αυτή μπορεί να γίνει μέσω GUI(Control Panel/Power Options) αλλά και μέσω του built-in command line tool `powercfg`.

Στο κεφάλαιο 3 θα δούμε τη βασική σύνταξη των εντολών του `powercfg` και τις διαφορές που εμφανίζει το εργαλείο στα Windows XP και στις νεότερες εκδόσεις.



Εικόνα 8: Ρυθμίσεις Ενέργειας των Windows

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό πραγματοποιήσαμε μια σύντομη ιστορική αναδρομή στα λειτουργικά συστήματα Microsoft Windows, έγινε ανάλυση της μέχρι σήμερα αρχιτεκτονικής τους (αρχιτεκτονική NT) και των βασικών συστατικών των Windows. Επίσης αναφερθήκαμε αναλυτικά στη δομή του μητρώου (registry) των Windows και τις λειτουργίες που επιτελεί, στην υποδομή WMI, στην τεχνολογία που χρησιμοποιούν οι διάφοροι Installers που υπάρχουν διαθέσιμοι στο λειτουργικό σύστημα Windows και τέλος παρουσιάσαμε μερικές από τις πιο κοινές δυσκολίες και προβλήματα στην εφαρμογή μεθόδων συντήρησης σε Windows συστήματα που καλείται να αντιμετωπίσει η εφαρμογή μας. Στο επόμενο κεφάλαιο θα αναφερθούμε σε μερικά από τα πιο σημαντικά εργαλεία διαχείρισης των Windows και θα επιδείξουμε βασικές λειτουργίες τους.

ΚΕΦΑΛΑΙΟ 2

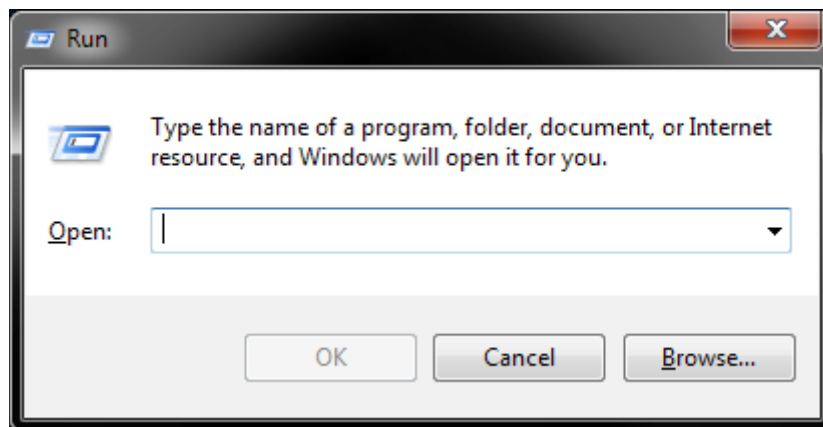
ΣΗΜΑΝΤΙΚΑ ΕΡΓΑΛΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ WINDOWS ΣΥΣΤΗΜΑΤΩΝ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα γίνει αναφορά στα πιο σημαντικά εργαλεία διαχείρισης των Windows, πως μπορεί να τα βρει ο χρήστης καθώς και μια επίδειξη/παρουσίαση των βασικών λειτουργιών τους.

2.1 Ο διάλογος Εκτέλεσης (Run Prompt)

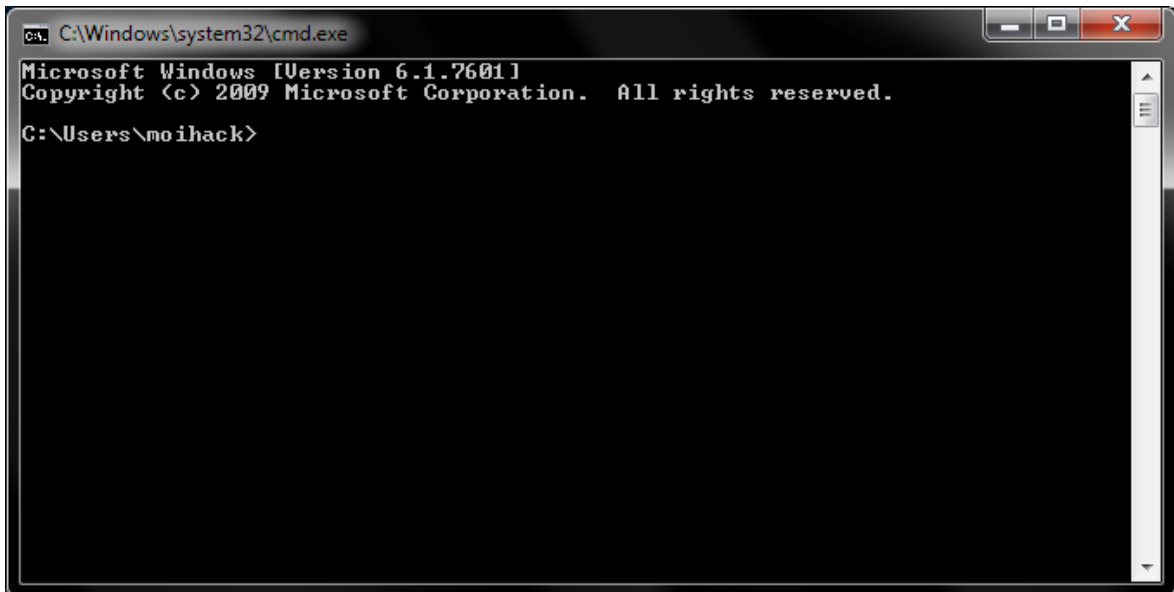
Ο διάλογος εκτέλεσης δεν αποτελεί ακριβώς ένα εργαλείο. Πρόκειται απλά για ένα διάλογο που εμφανίζεται όταν πιέσουμε ταυτόχρονα τα πλήκτρα Win και R (R από τη λέξη run – εκτέλεση) . Στο διάλογο αυτό μπορούμε να γράψουμε το όνομα του εκτελέσιμου που επιθυμούμε να εκτελέσουμε καθώς και τυχόν παραμέτρους που επιθυμούμε. Για οικονομία χώρου και χρόνου στο παρακάτω κείμενο, τα εργαλεία που θα αναλύσουμε θα εκτελούνται μέσω του run prompt για να αποφεύγονται οι αλληπάλληλες οδηγίες για την ξεχωριστή εύρεση του εκτελέσιμου του κάθε εργαλείου. Για αυτό το λόγο όταν υπάρχει κάποια πρόταση της μορφής “σε μία Run Prompt δίνουμε ...” αυτόματα ο χρήστης αν θέλει να αναπαράγει τα παραδείγματα της εργασίας μας πρέπει να χρησιμοποιεί το συνδυασμό πλήκτρων Win και R για την εμφάνιση της προτροπής εκτέλεσης.



Εικόνα 9: Ο διάλογος Εκτέλεσης (Run Prompt)

2.2 Η Γραμμή Εντολών (Command Prompt)

Η γραμμή εντολών (Command Prompt – για συντομία cmd) αποτελεί ένα πρόγραμμα των Windows (με όνομα εκτελέσιμου cmd.exe) . Για την εμφάνιση της δίνουμε σε μια Run Prompt τη γραμμή cmd.exe και πατάμε Enter. Θα εμφανιστεί ένα παράθυρο όπως αυτό της παρακάτω εικόνας:



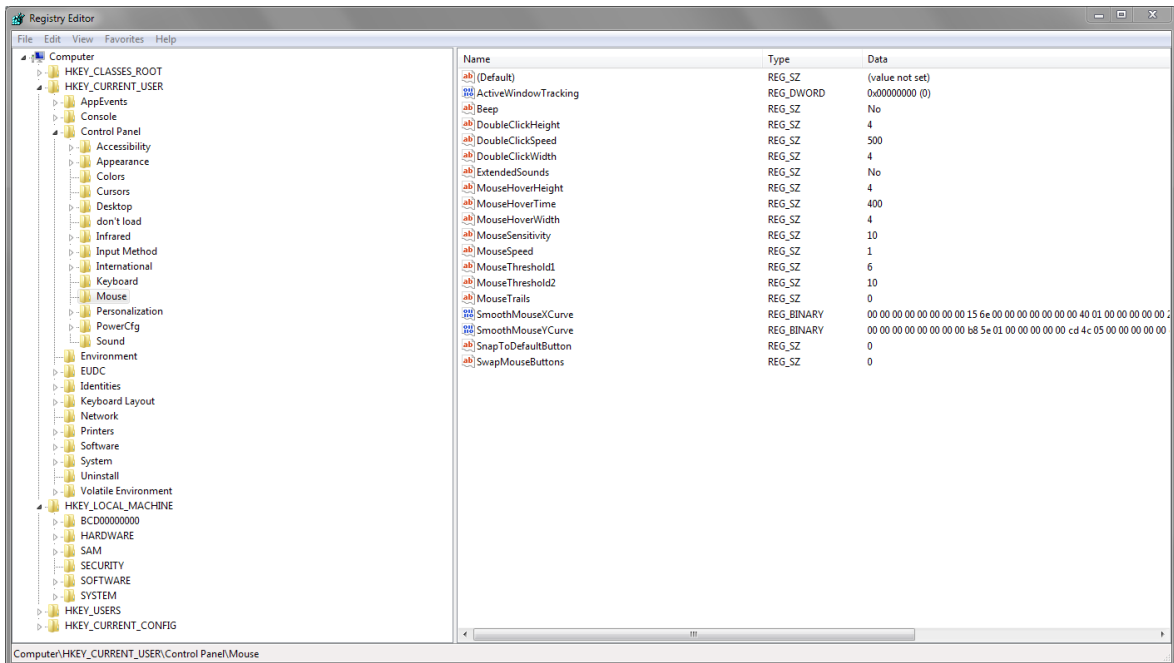
Εικόνα 10: Η Γραμμή Εντολών (Command Prompt)

Η γραμμή εντολών αποτελεί ότι και το τερματικό/κονσόλα σε ένα UNIX λειτουργικό. Αποτελεί δηλαδή μια πρόσβαση στο φλοιό(shell) του λειτουργικού συστήματος Windows και μας επιτρέπει την εκτέλεση εντολών τόσο εργαλείων κονσόλας όσο και προγραμμάτων με γραφικό περιβάλλον. Γενικά η γραμμή εντολών αποτελεί ένα ισχυρό εργαλείο που μας επιτρέπει να περιηγηθούμε και στο filesystem του συστήματος να δημιουργήσουμε αρχεία και φακέλους, ακόμα και να τρέξουμε μικρά προγράμματα εντολών (scripts). Αυτές οι λειτουργίες όμως ξεφεύγουν από το σκοπό του παρόντος κειμένου. Εμείς απλά θα κρατήσουμε την εικόνα της γραμμής εντολών καθώς και τη χρήση που έχει στην εφαρμογή μας, που δεν είναι άλλη παρά την κρυφή από το χρήστη εκτέλεσής της για την εκτέλεση άλλων ενεργειών.

2.3 Το εργαλείο regedit (Registry Editor)

Ο Επεξεργαστής Μητρώου (Registry Editor) ή regedit όπως είναι γνωστό το όνομα του εργαλείου στα Windows , αποτελεί ένα πρόγραμμα με γραφικό περιβάλλον που παρουσιάζει στο χρήστη την πλήρη τωρινή εικόνα του μητρώου του

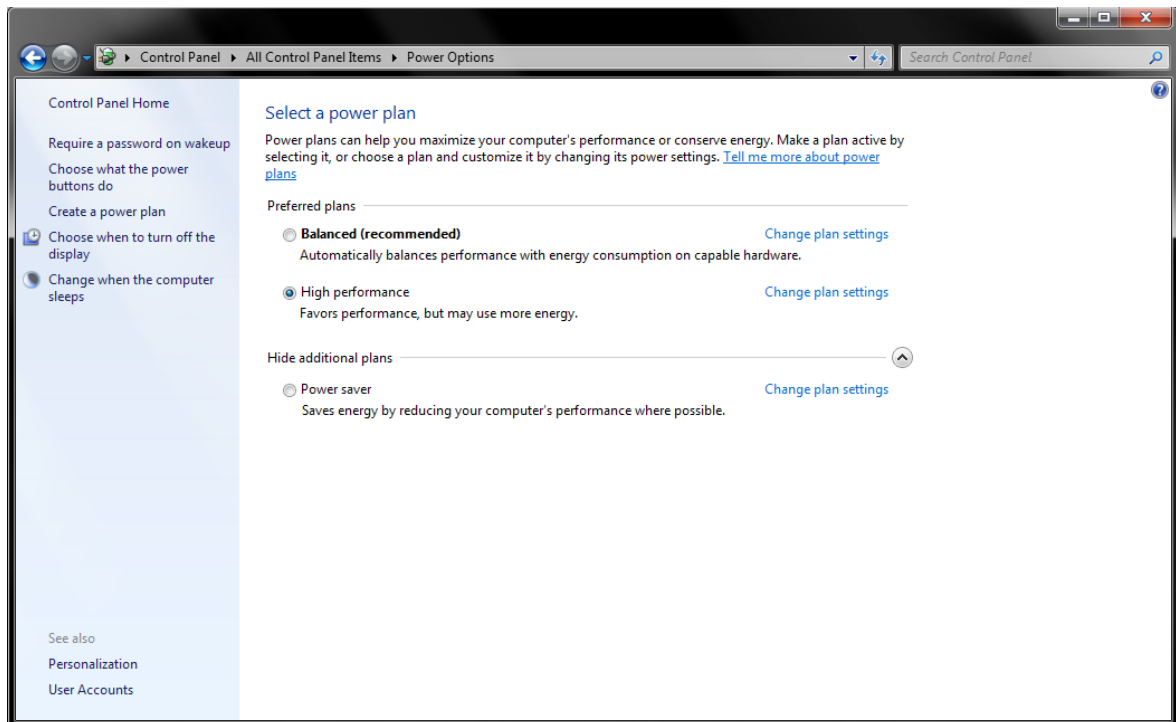
συστήματος. Ο regedit αποτέλεσε ένα σημαντικό εργαλείο κατά τη διάρκεια της συγγραφής κώδικα για τον AdvancedSystemManager καθώς επιτρέπει όχι μόνο την εύκολη αναζήτηση και προβολή κλειδιών και τιμών, αλλά και την δυνατότητα εξωτερικής αποθήκευσης αυτών γεγονός που διευκολύνει πολύ σε περιπτώσεις που έπρεπε να γίνουν δοκιμές σχετικές με διαγραφή κλειδιών. Το γραφικό περιβάλλον του regedit φαίνεται στην παρακάτω εικόνα:



Εικόνα 11: Το εργαλείο regedit

2.4 Το εργαλείο powercfg (Power Config)

Το εργαλείο powercfg αποτελεί την command line έκδοση του γραφικού περιβάλλοντος των Windows για τις ρυθμίσεις ενέργειας.



Εικόνα 12: Το γραφικό περιβάλλον ρυθμίσεων ενέργειας των Windows

Μέσα από το εργαλείο powercfg ο χρήστης μπορεί να ρυθμίσει και τη μέγιστη συχνότητα/ποσοστό λειτουργίας του επεξεργαστή. Δυστυχώς το εργαλείο powercfg δέχθηκε αλλαγές μετά την έκδοση των Windows XP οπότε είμαστε υποχρεωμένοι να αναφέρουμε και τυχόν διαφορές ανάμεσα στις εντολές και τον τρόπο σύνταξής τους.

Πίνακας 4: Οι εντολές του εργαλείου powercfg

ΠΑΡΑΜΕΤΡΟΣ	ΠΕΡΙΓΡΑΦΗ	Windows XP	Άλλα Windows
/QUERY	Δείχνει και αναλύει λεπτομερώς όλα τα power schemes του συστήματος.	ΝΑΙ	ΝΑΙ
/GETACTIVESCHEME	Εμφανίζει ποιο είναι το τρέχον power scheme του συστήματος.	ΟΧΙ	ΝΑΙ

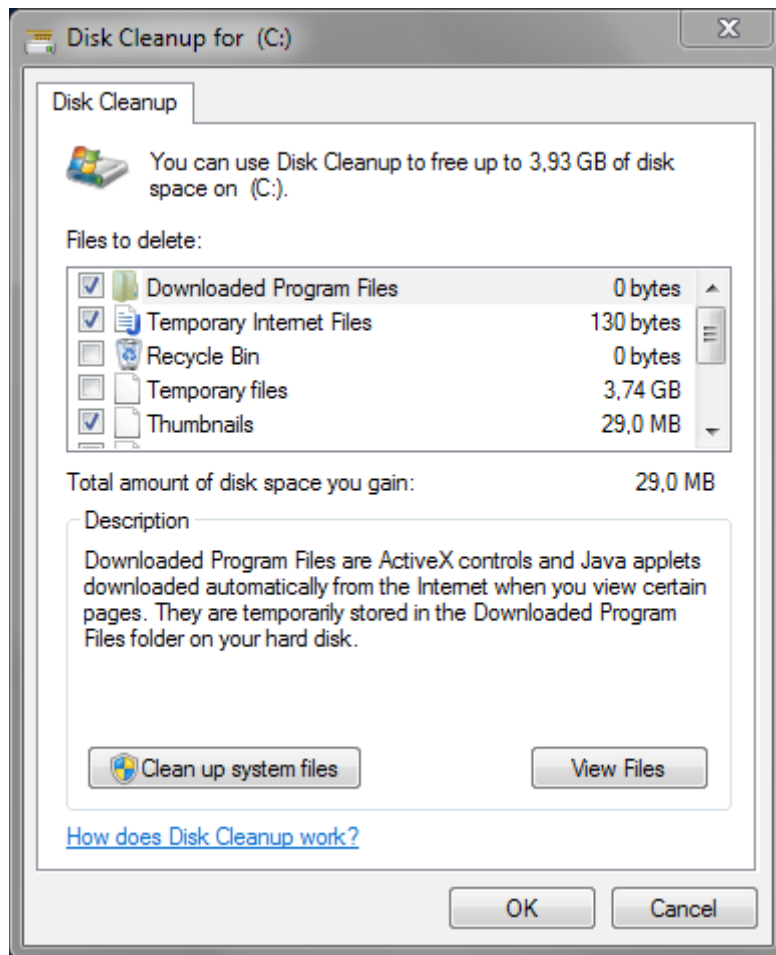
ΠΑΡΑΜΕΤΡΟΣ	ΠΕΡΙΓΡΑΦΗ	Windows XP	Άλλα Windows
/SETACVALUEINDEX	<p>Θέτει μια τιμή για το power scheme που ακολουθεί και θα χρησιμοποιείται όταν το σύστημα βρίσκεται συνδεδεμένο άμεσα στο ρεύμα.</p> <p>Συνδυάζεται με την παράμετρο “SUB_PROCESSOR PROCTHROTTLEMAX 100” για την εφαρμογή της μέγιστης συχνότητας λειτουργίας στον επεξεργαστή όσο βρίσκεται συνδεδεμένο στο ρεύμα.</p>	OXI	NAI
/SETDCVALUEINDEX	<p>Θέτει μια τιμή για το power scheme που ακολουθεί και θα χρησιμοποιείται όταν το σύστημα βρίσκεται συνδεδεμένο με μια μπαταρία</p> <p>Συνδυάζεται με την παράμετρο “SUB_PROCESSOR PROCTHROTTLEMAX 100” για την εφαρμογή της μέγιστης συχνότητας λειτουργίας στον επεξεργαστή όσο βρίσκεται συνδεδεμένο με μια μπαταρία.</p>	OXI	NAI
/SETACTIVE	<p>Εφαρμόζει τυχόν αλλαγές που πραγματοποιήθηκαν στο τρέχον power scheme. Διαφορετικά θα χρειαζόταν επανεκκίνηση για την εμφάνιση της επίδρασης των νέων ρυθμίσεων.</p>	NAI	NAI

Σημείωση: Στα Windows XP υπάρχει η παράμετρος “processor-throttle-ac throttle” και “processor-throttle-dc throttle” αντίστοιχα που πραγματοποιεί την λειτουργία των “/SETACVALUEINDEX” και “/SETDCVALUEINDEX” σε συνδυασμό με την “SUB_PROCESSOR PROCTHROTTLEMAX 100” . Είναι μία από τις πολλές αλλαγές που πραγματοποιήθηκαν στο εργαλείο powercfg. Ο

AdvancedSystemManager ελέγχει κατά την εκτέλεση την έκδοση του λειτουργικού που εκτελείται και ανάλογα τροποποιεί την εντολή που θα εκτελέσει.

2.5 Το εργαλείο *cleanmgr* (Disk Cleanup)

Το εργαλείο *cleanmgr* αναλαμβάνει να “καθαρίσει” το σκληρό δίσκο του συστήματος από περιττά αρχεία εγκαταστάσεων, αρχεία αποσφαλμάτωσης, προσωρινά αρχεία και άλλα. Για να το εκτελέσει ο χρήστης αρκεί να δώσει σε ένα Run Prompt “*cleanmgr*” χωρίς τα εισαγωγικά και να πατήσει Enter. Έπειτα από ένα (πιθανό) διάλογο προτροπής για επιλογή σκληρού δίσκου θα εμφανιστεί στο χρήστη η αντίστοιχη διεπιφάνεια χρήστη της εικόνας.



Εικόνα 13: Το γραφικό περιβάλλον του εργαλείου *Disk Cleanup*

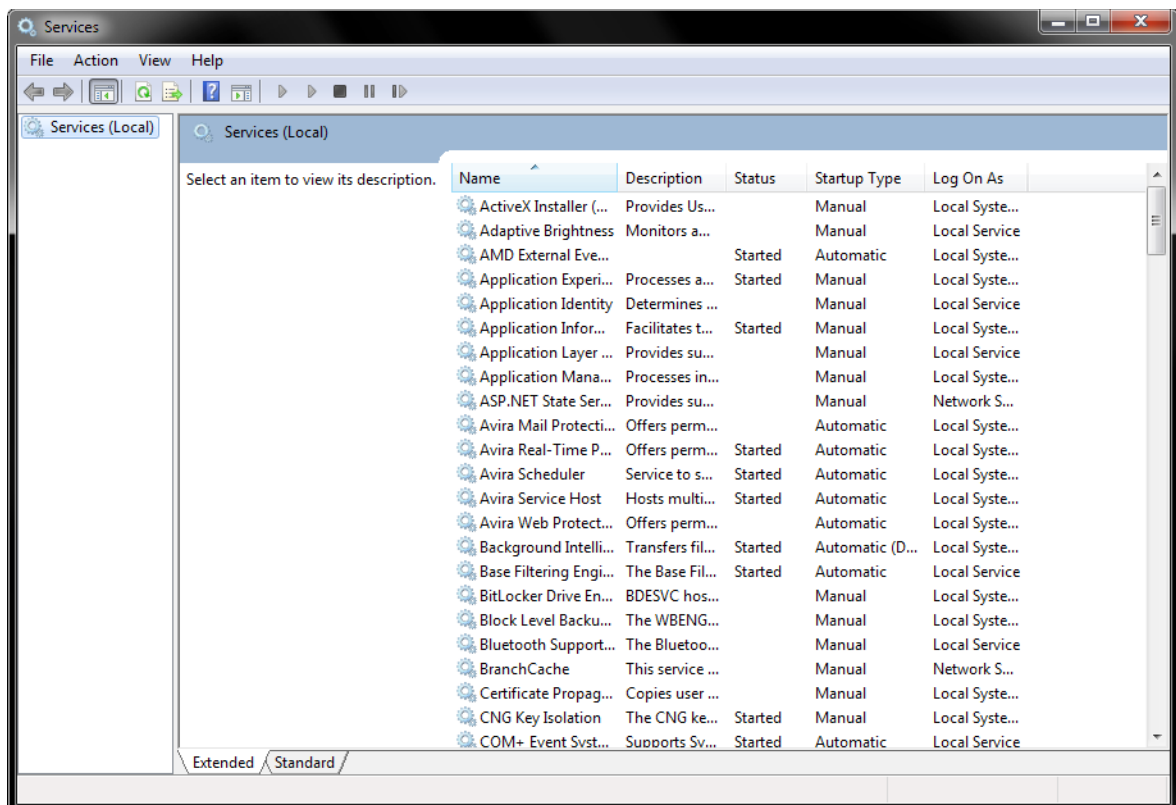
Ο χρήστης μπορεί να επιλέξει τις επιλογές που θέλει και πατώντας το κουμπί “OK” το εργαλείο *Disk Cleanup* διαγράφει από το δίσκο τα αρχεία που ταιριάζουν

ανάλογα με τις επιλογές του χρήστη. Δυστυχώς το εργαλείο Disk Cleanup δεν προσφέρει ένα command line περιβάλλον όπως το powercfg για τις ρυθμίσεις ενέργειας. Έτσι ανακατασκευάσαμε τη λειτουργία του στην κλάση DiskCleanup του AdvancedSystemManager.

Σημείωση: Τόσο η εφαρμογή Disk Cleanup των Windows όσο και ο AdvancedSystemManager δεν διαγράφουν αρχεία του χρήστη. Τα αρχεία που εντοπίζονται και διαγράφονται με αυτές τις μεθόδους δεν αποτελούν κοινούς καταλόγους που ένας χρήστης αποθηκεύει τα αρχεία του.

2.6 Το εργαλείο *services.msc* (Services)

Το εργαλείο Services αποτελεί το εργαλείο με γραφικό περιβάλλον που προσφέρουν τα Windows με σκοπό τη διαχείριση των υπηρεσιών (services) που εκτελούνται στο σύστημα. Το γραφικό περιβάλλον που παρέχεται φαίνεται στην παρακάτω εικόνα.



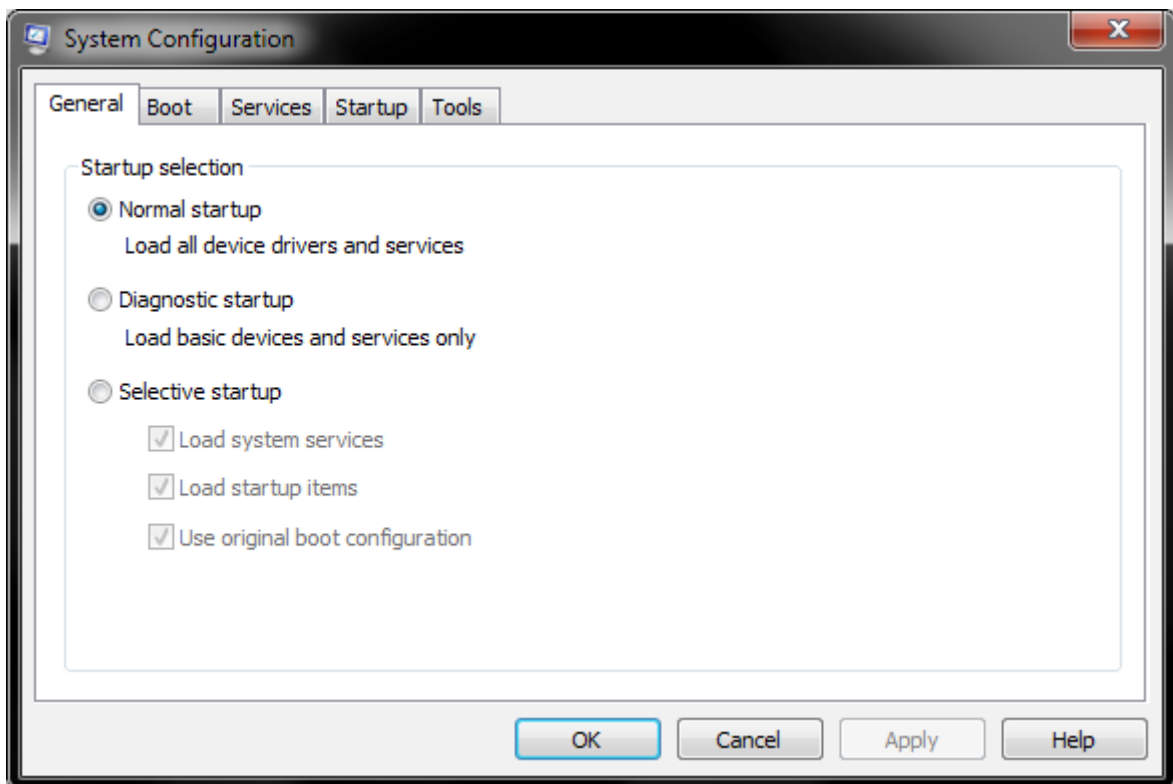
Εικόνα 14: Το γραφικό περιβάλλον του εργαλείου *services*

Στο περιβάλλον αυτό παρουσιάζονται σε μορφή λίστας όλες οι υπηρεσίες που εκτελούνται στο σύστημα μαζί με επιπλέον πληροφορίες που περιλαμβάνουν την

περιγραφή των λειτουργιών αυτών των υπηρεσιών, την κατάστασή τους (αν βρίσκονται σε εκτέλεση ή όχι) , τον τύπο εκκίνησης τους (αυτόματη, χειροκίνητη, αυτόματη με καθυστέρηση, απενεργοποιημένη) και το χρήστη που τις εκτελεί. Για την πρόσβαση στο γραφικό περιβάλλον Services ο χρήστης μπορεί να δώσει σε μια Run Prompt “services.msc” χωρίς τα εισαγωγικά και να πατήσει Enter.

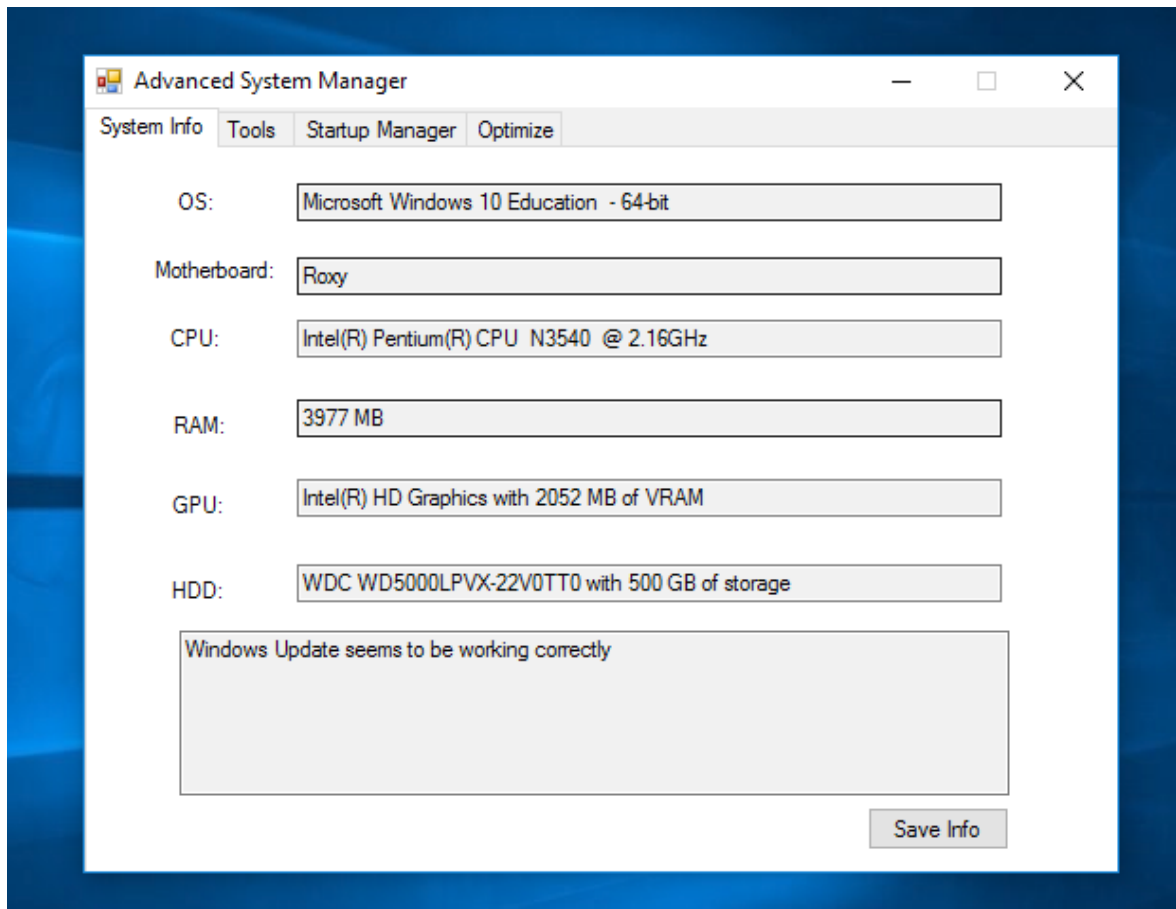
2.7 Το εργαλείο msconfig

Το εργαλείο msconfig είναι ίσως από τα πιο σημαντικά εργαλεία διαχείρισης των Windows. Ακόμα και αυτό όμως παραμένει κρυφό από το χρήστη. Για την εμφάνισή του ο χρήστης πρέπει να δώσει σε μία Run Prompt “msconfig” χωρίς τα εισαγωγικά και να πατήσει Enter. Θα εμφανιστεί μια διεπιφάνεια χρήστη παρόμοια με αυτή της εικόνας:



Εικόνα 15: Το εργαλείο msconfig

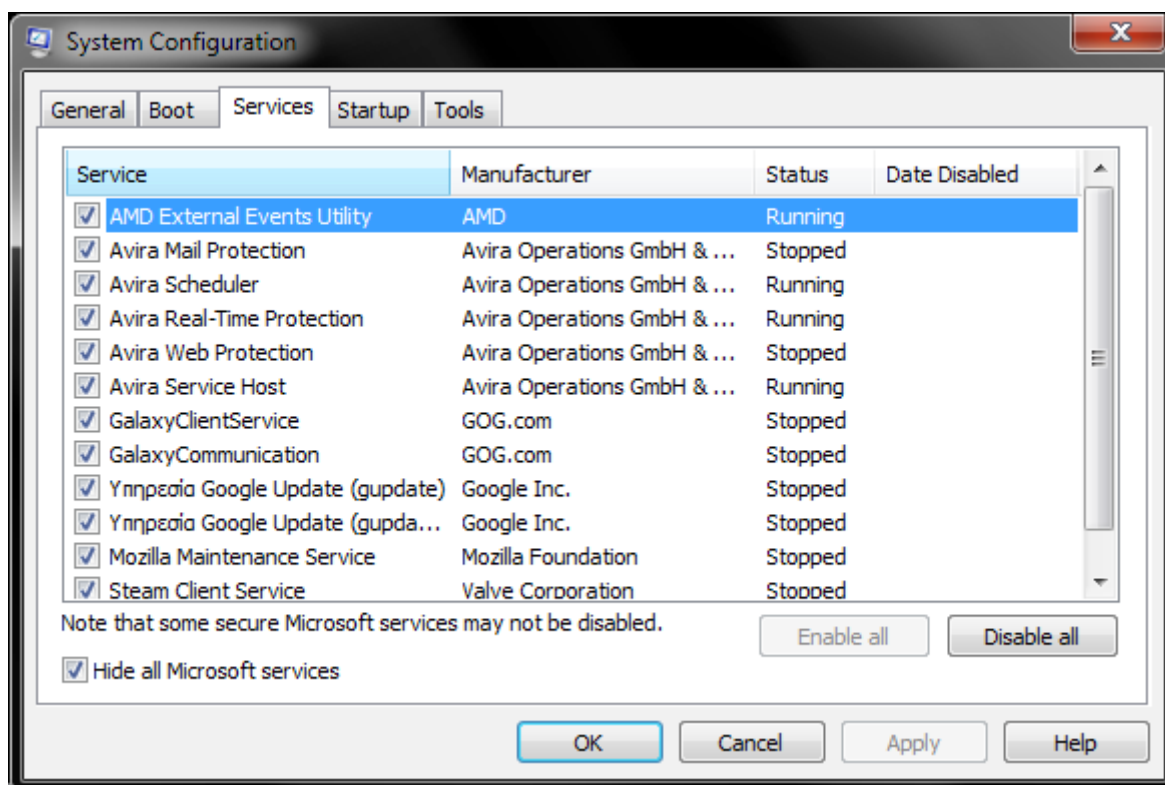
Το περιβάλλον του msconfig και οι λειτουργίες του επιρρέασαν σε μεγάλο βαθμό το περιβάλλον και την οργάνωση της διεπιφάνειας του AdvancedSystemManager, κάτι που είναι εμφανές με μια απλή σύγκριση των αρχικών οθονών των δύο προγραμμάτων.



Εικόνα 16: Σύγκριση με AdvancedSystemManager

Υπάρχει μια κοινή οργάνωση της διεπιφάνειας χρήστη χρησιμοποιώντας καρτέλες(tabs) και ήδη μπορούμε να δούμε μερικές κοινές ονομασίες όπως (Tools, Startup → Startup Manager) .

Το περιβάλλον όμως του AdvancedSystemManager θα αναλυθεί εκτενέστερα στον οδηγό χρήσης λογισμικού στο τέλος αυτής της εργασίας. Ας συνεχίσουμε με την περιγραφή των λειτουργιών του εργαλείου msconfig.



Εικόνα 17: Η καρτέλα διαχείρισης υπηρεσιών του εργαλείου msconfig

Η καρτέλα Services στο msconfig παρουσιάζει στον χρήστη σε μορφή λίστας τον κατάλογο των υπηρεσιών που εκκινούν αυτόματα κατά την εκκίνηση του συστήματος. Όσες έχουν το χαρακτηριστικό “τσεκ” μπροστά σημαίνει πως εκκινούν αυτόματα ενώ οι υπόλοιπες όχι. Στη στήλη Manufacturer(Κατασκευαστής) ο χρήστης μπορεί να δει τον εκδότη της υπηρεσίας που συνήθως είναι το όνομα της εταιρείας που κατασκεύασε το λογισμικό της υπηρεσίας. Στη στήλη Status(Κατάσταση) φαίνεται το στάδιο εκτέλεσης της κάθε υπηρεσίας με πιθανές τιμές “Running”(“Σε εκτέλεση”) και “Stopped”(“Σταματημένη”). Στη στήλη “Date Disabled” υπάρχει (όχι απαραίτητα) μία ημερομηνία που μια υπηρεσία καταργήθηκε από την αυτόματη εκκίνηση. Στο κάτω αριστερό μέρος της καρτέλας υπάρχει η επιλογή “Hide all Microsoft services” όπου αν επιλεγεί από το χρήστη, τότε αποκρύπτονται από τη λίστα(προσωρινά – για όσο θα είναι ενεργή η επιλογή) οι υπηρεσίες του ίδιου του λειτουργικού συστήματος και εμφανίζονται μόνο αυτές που εγκαταστάθηκαν στο σύστημα από προγράμματα που εγκατέστησε ο ίδιος ο χρήστης. Αυτό είναι ιδιαίτερα χρήσιμο σε περίπτωση που ο χρήστης θέλει να καταργήσει κάποιες υπηρεσίες από το να εκκινούν αυτόματα και ταυτόχρονα να ναι σίγουρος πως δεν θα προκληθεί ζημιά στο γενικότερο σύστημα. Ο

AdvancedSystemManager εμφανίζει μόνο τις υπηρεσίες που δεν αποτελούν μέρος του συστήματος για τροποποίηση από τον χρήστη.

Σημείωση: Η κατάσταση(status) είναι κάτι τελείως διαφορετικό και ανεξάρτητο από το αν μια υπηρεσία εκκινεί αυτόματα στην εκκίνηση. Μια υπηρεσία μπορεί να εκκινεί αυτόματα και παρόλα αυτά η κατάστασή της να είναι σταματημένη (stopped). Αυτό μπορεί να συμβεί για παράδειγμα σε μια υπηρεσία ελέγχου ενημερώσεων για ένα πρόγραμμα. Στην εκκίνηση ελέγχεται αν υπάρχει μια νέα έκδοση προγράμματος και αν δεν βρεθεί τότε η υπηρεσία σταματά να εκτελείται μέχρι την επομένη εκκίνηση του υπολογιστή.

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό παρουσιάσαμε τα πιο σημαντικά εργαλεία για τη διαχείριση ενός Windows συστήματος και αναλύσαμε τη λειτουργικότητά τους. Επίσης αναφέραμε τον εύκολο τρόπο πρόσβασης σε αυτά μέσω της προτροπής εκτέλεσης (Run Prompt). Στο επόμενο κεφάλαιο θα πραγματοποιήσουμε μια εισαγωγή στο προγραμματιστικό περιβάλλον .NET Framework της Microsoft και θα αναλύσουμε μερικές από τις σημαντικές βιβλιοθήκες που χρησιμοποιήσαμε στην εφαρμογή μας.

ΚΕΦΑΛΑΙΟ 3

ΕΙΣΑΓΩΓΗ ΣΤΟ MICROSOFT .NET FRAMEWORK

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα κάνουμε μια εισαγωγή στο πλαίσιο λογισμικού της Microsoft με την ονομασία .NET Framework. Πιο συγκεκριμένα θα αναλύσουμε την δομή και τον τρόπο λειτουργίας του και θα αναφέρουμε τις πιο σημαντικές κλάσεις και βιβλιοθήκες του που χρησιμοποιούνται στην εφαρμογή μας.

3.1 Εισαγωγή στο .NET Framework

Το .NET Framework (προφέρεται dot net) είναι ένα framework(πλαίσιο) λογισμικού που αναπτύσσεται από την Microsoft και τρέχει κυρίως σε Windows συστήματα. Το .NET περιλαμβάνει μια μεγάλη βιβλιοθήκη κλάσεων την Framework Class Library (FCL). Η χρήση του .NET Framework επιτρέπει την λεγόμενη “Language Interoperability”(κάθε γλώσσα του .NET μπορεί να χρησιμοποιήσει κώδικα που γράφτηκε σε άλλη γλώσσα του .NET). Το .NET δεν τρέχει σε περιβάλλον hardware όπως μια compiled γλώσσα όπως η C++ αλλά σε ένα εικονικό software περιβάλλον το Common Language Runtime(CLR). Το CLR πρόκειται για μια εικονική μηχανή (σαν την Java Virtual Machine-JVM) και παρέχει τις απαραίτητες υπηρεσίες όπως αυτές της διαχείρισης μνήμης και του χειρισμού εξαιρέσεων. Κώδικας που γράφτηκε χρησιμοποιώντας το .NET Framework αναφέρεται στη βιβλιογραφία ως “Managed Code” ενώ αντίστοιχα κώδικας όπως ενός προγράμματος της C++ αναφέρεται ως “Un-Managed Code”. Η FCL και το CLR αποτελούν μαζί το .NET Framework.

Η FCL περιλαμβάνει βιβλιοθήκες για πληθώρα λειτουργιών όπως η δημιουργία διεπαφών χρήστη, πρόσβαση σε κρυπτογραφικές συναρτήσεις, δημιουργία και χρήση network sockets και άλλες. Το Visual Studio αποτελεί ένα IDE(Integrated Development Environment) που αναπτύσσει η Microsoft για ανάπτυξη λογισμικού που χρησιμοποιεί το .NET Framework . Αν και υπάρχουν και άλλα IDE (για παράδειγμα το MonoDevelop) ,το Visual Studio έχει επικρατήσει για τις ευκολίες που παρέχει στον προγραμματιστή καθώς και τα διάφορα εργαλεία ανάλυσης και μετρικών που περιλαμβάνει.

Το .NET Framework πρωτοεμφανίστηκε το 1990 με πολλές εκδόσεις να ακολουθούν και ενσωματώνοντας ολοένα και περισσότερες καινούριες βιβλιοθήκες για τους προγραμματιστές. Όμως με τις νέες εκδόσεις εμφανίζεται και η ανάγκη για ένα νέο runtime component στο σύστημα. Έτσι εκδόσεις που γράφτηκαν σε

νεότερο .NET Framework είναι πιθανό (ανάλογα με το είδος των βιβλιοθηκών και μεθόδων που χρησιμοποιούνται) να μην μπορούν να εκτελεστούν ή να συμβαίνουν exceptions λόγω της ασυμβατότητας του runtime του εκτελέσιμου και αυτού που υπάρχει στο σύστημα. Στον πίνακα της εικόνας φαίνονται οι διαφορετικές εκδόσεις του .NET Framework και η ημερομηνία κυκλοφορίας τους. Για την ανάπτυξη του προγράμματος της εργασίας μας, επιλέχθηκε το .NET Framework 2.0 για μεγιστοποίηση της συμβατότητας με τις περισσότερες εκδόσεις των Windows χωρίς να χρειάζεται κάποια επιπλέον εγκατάσταση του .NET runtime. Ακόμα και μηχανήματα με Windows XP έχουν στις περισσότερες περιπτώσεις έκδοση ίση ή μεγαλύτερη του 2.0. Το .NET Framework είναι backwards compatible, δηλαδή εφαρμογές που έχουν γραφτεί με μία παλιότερη έκδοση του .NET Framework μπορούν να εκτελεστούν κανονικά αν υπάρχει στο σύστημα νεότερη έκδοση του runtime.

3.2 Σημαντικά συστατικά του .NET Framework

Windows Forms

Οι Windows Forms (Windows Φόρμες) αποτελούν μια βιβλιοθήκη γραφικών του .NET Framework για τη δημιουργία Γραφικών Διεπαφών Χρήστη (GUI).

Μια εφαρμογή γραμμένη σε Windows Forms είναι μια εφαρμογή καθοδηγούμενη από γεγονότα (event driven). Δηλαδή η ροή εκτέλεσης της εφαρμογής δεν προχωράει αν δεν συμβεί μια ενέργεια από το χρήστη όπως το πάτημα ενός κουμπιού. Σε μια εφαρμογή Windows Forms ο προγραμματιστής τοποθετεί User Controls, στοιχεία δηλαδή όπως κουμπιά, textboxes, labels και άλλα δημιουργώντας τελικά το γραφικό περιβάλλον της εφαρμογής.

Background Worker

Όταν υπάρχει μια διαδικασία η οποία θα πάρει αρκετή ώρα για να ολοκληρωθεί και δεν θέλουμε να προκαλέσει καθυστερήσεις στο περιβάλλον και την εκτέλεση της εφαρμογής μας τότε μπορούμε να χρησιμοποιήσουμε την κλάση BackgroundWorker για να εκτελέσουμε την συγκεκριμένη διαδικασία σε ένα άλλο thread (νήμα) από αυτό που τρέχει το γραφικό περιβάλλον της εφαρμογής μας χωρίς να χρειαστεί να χρησιμοποιήσουμε κλάσεις της System.Threading και να ασχοληθούμε ιδιαίτερα με τον συγχρονισμό νημάτων. [7,8]

Το Visual Studio μας παρέχει την BackgroundWorker σαν component στο Toolbox.

Ακριβώς όπως βάζουμε ένα Textbox στη φόρμα μας τοποθετούμε με τον ίδιο τρόπο και έναν BackgroundWorker. Μόλις το κάνουμε αυτό στο παράθυρο

properties του BackgroundWorker(bgw1 πλέον) μπορούμε να μελετήσουμε τα διάφορα events του.

Αυτά είναι τα παρακάτω:

DoWork : συμβαίνει όταν καλείται η RunWorkerAsync μέθοδος του worker μας. Εδώ ο bgw1 ξεκινά και εκτελεί την μακροχρόνια διαδικασία που θέλουμε.

ProgressChanged: συμβαίνει όταν καλείται η ReportProgress μέθοδος του worker. Εδώ συμβαίνουν διάφορες διαδικασίες ενημέρωσης(συνήθως το γέμισμα ενός progress bar) στοιχείων της διεπιφάνειας ώστε να αντιλαμβάνεται ο χρήστης την πρόοδο της χρονικά μεγάλης διαδικασίας που εκτελεί ο bgw1. Προσοχή! Για να μπορεί να χρησιμοποιηθεί η ReportProgress(και αντίστοιχα να συμβεί και το event ProgressChanged) πρέπει να έχουμε ορίσει στις ιδιότητες του bgw1 ότι ReportsProgress = true, διαφορετικά θα έχουμε exception.

RunWorkerCompleted: συμβαίνει όταν τελειώνει η εκτέλεση της διαδικασίας που εκτελείται στο παρασκήνιο. Το event εκτελείται πάντα δηλαδή ακόμα και όταν η διαδικασία στο παρασκήνιο αποτύχει με exception είτε απλά ο χρήστης ακυρώσει την διαδικασία.

Σημείωση: Ένας BackgroundWorker δεν μπορεί να έχει πρόσβαση/τροποποιεί στοιχεία του γραφικού περιβάλλοντος της εφαρμογής μας καθώς θα προκληθεί exception με μήνυμα "Cross-thread operation not valid". Παρ'όλα αυτά στα events OnProgressChanged και OnRunWorkerCompleted ο έλεγχος του προγράμματος επιστρέφει στο interface μας και επομένως από εκεί να αλλάξουμε/τροποποιήσουμε στοιχεία του γραφικού περιβάλλοντος της εφαρμογής μας(πχ μια progress bar ή ένα label). Από το DoWork μπορούμε όμως να έχουμε πρόσβαση ή ακόμα και να τροποποιήσουμε static μεταβλητές. Αν θέλουμε όμως να αποφύγουμε τα προβλήματα συγχρονισμού πρέπει να είμαστε σίγουροι ότι αυτές οι static μεταβλητές δεν τροποποιούνται από άλλα νήματα την ώρα που εκτελείται η διεργασία μας στον bgw1.

Η κλάση Process

Η κλάση Process μας δίνει πρόσβαση σε τοπικές και απομακρυσμένες διαδικασίες(processes) και μας επιτρέπει να ξεκινήσουμε νέες καθώς και να σταματήσουμε τις ήδη υπάρχουσες που εκτελούνται. Η κλάση Process χρησιμοποιείται σε μεγάλο βαθμό στο project μας για την εκτέλεση μερικών ενσωματωμένων εργαλείων των Windows περνώντας τους παράλληλα ειδικές παραμέτρους. Για αυτό το λόγο εδώ θα παρουσιάσουμε μερικές από τις ιδιότητες της και πως αυτές χρησιμοποιούνται στο project μας.

Αρχικά χρειάζεται να εισάγουμε τη βιβλιοθήκη Diagnostics για να μπορούμε να χρησιμοποιήσουμε την κλάση Process και τις μεθόδους της. Για να εισάγουμε τη βιβλιοθήκη απλά στην κορυφή του αρχείου μας προσθέτουμε: “using System.Diagnostics;” χωρίς τα εισαγωγικά.

Η κλάση Process συνδέεται αρκετά με την κλάση ProcessStartInfo μία κλάση που πάλι βρίσκεται στο namespace System.Diagnostics. Κάνοντας χρήση της κλάσης ProcessStartInfo μπορούμε να δημιουργήσουμε ένα αντικείμενο το οποίο περιέχει όλες εκείνες τις παραμέτρους(arguments) που ενδεχομένως χρειάζεται κάποιος να “περάσει” σε ένα πρόγραμμα. Οι παράμετροι αυτοί περιλαμβάνουν αρκετές σημαντικές λειτουργίες όπως:

CreateNoWindow: όπως μαρτυρά και το όνομα της καθορίζει αν το Process που θα ξεκινήσει, αν θα δημιουργήσει ένα καινούριο παράθυρο στην οθόνη. By default, έχει τιμή false δηλαδή δεν δημιουργείται νέο παράθυρο.

UseShellExecute: Όταν είναι true χρησιμοποιείται το κέλυφος(shell) του συστήματος για να εκτελεστεί/ξεκινήσει το process. Διαφορετικά αν είναι false εκτελείται απευθείας το εκτελέσιμο αρχείο που επιθυμούμε. By default είναι true.

RedirectStandardError: Ανακατεύθυνση του Standard Error Stream του εκτελέσιμου που πρόκειται να καλέσουμε με την κλάση Process. Με την ανακατεύθυνση αποκτά πρόσβαση στο error stream και η εφαρμογή μας και αντίστοιχα μπορούμε να ανακατευθύνουμε την ροή σφάλματος εκτός από την κονσόλα αλλά και σε κάποιο αρχείο.

RedirectStandardOutput: Όμοια με το RedirectStandardError απλά για την ροή εξόδου του εκτελέσιμου.

Σημείωση: Για να μπορούν να ισχύουν οι τιμές που των RedirectStandardError και RedirectStandardOutput πρέπει το πεδίο UseShellExecute να έχει τιμή false!

WindowStyle: Εδώ καθορίζουμε αν το παράθυρο που θα δημιουργηθεί θα είναι ελαχιστοποιημένο,μεγιστοποιημένο,κανονικό ή αν δεν θα εμφανίζεται καθόλου (χρήσιμο όταν δεν είναι δυνατή η χρήση της CreateNoWindow)

Arguments: Εδώ συντάσσουμε ένα string με όλες τις παραμέτρους που επιθυμούμε να τρέξει το πρόγραμμα μας. Ιδιαίτερη προσοχή στα κενά ανάμεσα στις παραμέτρους καθώς και στη χρήση εισαγωγικών (“ ”) όταν χρειάζεται ένα path που περιλαμβάνει κενά σαν παράμετρο.

Έχοντας πλέον δώσει τις κατάλληλες τιμές στα πεδία της `ProcessStartInfo` περνάμε το αντικείμενο που δημιουργήσαμε σαν παράμετρο στο δομητή της `Process` και τελικά ξεκινάμε την εκτέλεση μας με κλήση της `Process.Start()`.

Σημείωση: Αν έχουμε ρυθμίσει το `Process` μας να μην εμφανίσει κάποιο παράθυρο, φαινομενικά αυτό από την οπτική μεριά του χρήστη αποτελεί μια εργασία που εκτελείται στο παρασκήνιο. Θα πρέπει όμως να διευκρινίσουμε εδώ ότι ένα `Process` που δεν “φαίνεται” το παράθυρο εκτέλεσης του στο χρήστη δεν σημαίνει πως δεν θα εμποδίσει τη συνέχεια εκτέλεσης του νήματος της εφαρμογής μας μέχρις ότου ολοκληρωθεί η εκτέλεση του. Αν δεν επιθυμούμε να συμβεί αυτό και θέλουμε πραγματικά να γίνει η εκτέλεση του `Process` στο παρασκήνιο (σε άλλο νήμα) χωρίς να εμποδίσει την συνέχεια της ροής εκτέλεσης της εφαρμογής μας, τότε θα χρειαστεί να χρησιμοποιήσουμε είτε την κλάση `Thread` είτε έναν `BackgroundWorker`.

ΕΠΙΛΟΓΟΣ

Σε αυτό το κεφάλαιο πραγματοποιήσαμε μια εισαγωγή στο `.NET Framework`, αναλύσαμε τη δομή του και αναφερθήκαμε στα σημαντικά συστατικά του που χρησιμοποιήθηκαν στην εργασία μας καλύπτοντας αναλυτικά τις μεθόδους και τον τρόπο λειτουργίας τους. Στο επόμενο κεφάλαιο θα αναλύσουμε σε βάθος τη δομή του κώδικα της εφαρμογής μας και πως σχεδιάστηκε το πρόγραμμα κάνοντας χρήση των προηγούμενων κεφαλαίων.

ΚΕΦΑΛΑΙΟ 4

ADVANCED SYSTEM MANAGER – ΑΝΑΛΥΣΗ ΟΝΤΟΤΗΤΩΝ

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα αναλύσουμε συνοπτικά μερικές από τις κλάσεις του `AdvancedSystemManager`, θα αναφερθούμε στη λειτουργία τους μέσα στο πρόγραμμα και στον τρόπο υλοποίησης των μεθόδων τους. Για την κατανόηση αυτού του κεφαλαίου καλό είναι ο αναγνώστης να έχει βασικές γνώσεις των εννοιών του αντικειμενοστρεφούς προγραμματισμού καθώς και να έχει μελετήσει προηγουμένως τα κεφάλαια 1 και 2 ώστε να έχει γνώση της πλατφόρμας `.NET` και της αρχιτεκτονικής των Windows συστημάτων. Τα παρακάτω αποτελούν περισσότερο ένα μικρό `documentation` – περιγραφή του προγράμματος και μια αναφορά του που υλοποιείται ποια διαδικασία, ώστε να βοηθηθούν μελλοντικοί προγραμματιστές που επιθυμούν τη βελτίωση του `AdvancedSystemManager`.

Η κλάση `MyLogger`

Θα ξεκινήσουμε με μια πολύ απλή κλάση, την κλάση `MyLogger`. Η κλάση αυτή περιέχει μονάχα `static` μεθόδους που χρησιμοποιούνται κυρίως για την καταγραφή των σφαλμάτων που τυχόν συμβαίνουν κατά την εκτέλεση της εφαρμογής. Έτσι τόσο ο χρήστης όσο και ένας προγραμματιστής του `AdvancedSystemManager` μπορεί να χρησιμοποιήσει αυτές τις μεθόδους για `debugging` και μηνύματα τα οποία αποθηκεύονται σε απλά αρχεία `.txt`. Έτσι σε περίπτωση σφάλματος ένας χρήστης μπορεί να παρέχει στον προγραμματιστή πληροφορίες απαραίτητες για την επίλυση του. Τα αρχεία που χρησιμοποιούνται για αυτό το σκοπό είναι τα `log.txt` και `error_log.txt` και δημιουργούνται αυτόματα. Αν υπάρχουν ήδη τότε σε κάθε νέα εκκίνηση του προγράμματος το περιεχόμενό τους χάνεται. Επίσης η κλάση `MyLogger` χρησιμοποιείται για να αποθηκεύσει την λίστα προγραμμάτων και τις πληροφορίες συστήματος σε αρχεία όταν αυτό ζητηθεί από το χρήστη.

Η κλάση `ArgumentParser`

Και εδώ πάλι έχουμε μια αρκετά απλή κλάση που χρησιμεύει όπως μαρτυρά και το όνομά της στην αναγνώριση των παραμέτρων που περνούν στο πρόγραμμα κατά την εκτέλεση του στο περιβάλλον κονσόλας των Windows. Η μέθοδος `ParseArguments` της `ArgumentParser` καλείται από το `Program.cs` που περιέχει την κλάση `Main` του προγράμματός μας. Εκεί ανάλογα τις παραμέτρους που δίνονται εκτελείται είτε το γραφικό περιβάλλον του `AdvancedSystemManager` είτε εκτελείται σε περιβάλλον κονσόλας.

Η κλάση SystemInfo

Αυτή η κλάση περιέχει ένα πλήθος πεδίων String που αποκτούν τιμές μέσω των κλήσεων που γίνονται στο **WMI** από το πρόγραμμα. Πιο συγκεκριμένα μέσα στο δομητή της κλάσης SystemInfo υπάρχουν πολλαπλές κλήσεις στο WMI, μία για κάθε χαρακτηριστικό/ιδιότητα της κλάσης. Η κλάση SystemInfo περιέχει τις πληροφορίες σχετικά με το hardware που υπάρχει στον υπολογιστή που εκτελείται το πρόγραμμα καθώς και την έκδοση των Windows που υπάρχει εγκατεστημένη. Περιέχει επίσης μια μέθοδο ToString για αναπαράσταση αυτής της πληροφορίας σε μορφή συμβολοσειράς.

```
using System.Management;

String propertyName;
String wmiClass;

ManagementObjectSearcher searcher = new ManagementObjectSearcher("SELECT " + propertyName +
" FROM " + wmiClass);

    foreach (ManagementObject mo in searcher.Get())
    {
        foreach (PropertyData property in mo.Properties)
        {
            Console.WriteLine(property.Value);
        }
    }
```

Εικόνα 18: Χρήση της WMI από τη C# - Παράδειγμα WQL ερωτήματος

Όπως φαίνεται στη κλήση της μεθόδου της WMI ManagementObjectSearcher έχουμε ένα απλό SQL/WQL ερώτημα που είναι κατανοητό σε οποιονδήποτε έχει ασχοληθεί με σχεσιακά συστήματα βάσεων δεδομένων. Για το πρόγραμμά μας ένα SQL/WQL query θα μπορούσε να είναι:

SELECT Name FROM Win32_VideoController δηλαδή εντόπισε το πεδίο Name(όνομα) στην κλάση της κάρτας γραφικών.

Χτίζοντας πάνω σε αυτό το βασικό query ο δομητής της SystemInfo αποκτά όλες τις πληροφορίες για το hardware του υπολογιστή που θα τρέχει το πρόγραμμά μας.

Η κλάση CPUPower

Μια ιδιαίτερα απλή κλάση. Απλά ελέγχει την έκδοση του λειτουργικού συστήματος που χρησιμοποιείται και εκτελεί στο background το εργαλείο **powercfg** των Windows με τις κατάλληλες παραμέτρους για την εφαρμογή της μέγιστης δυνατής συχνότητας λειτουργίας του επεξεργαστή.

Η κλάση WindowsService

Μια κλάση με 4 μόνο πεδία και ένα δομητή για την ακριβή αναπαράσταση μιας οντότητας τύπου Windows Service. Η καταμέτρηση όλων των Services γίνεται στην κλάση ServiceManager

Η κλάση Service Manager

Όπως προαναφέρθηκε η κλάση ServiceManager καταγράφει όλες τις υπηρεσίες που υπάρχουν εγκατεστημένες στο σύστημα. Η καταγραφή γίνεται σε μια static λίστα που δέχεται αντικείμενα τύπου WindowsService. Αρχικά χρησιμοποιείται η κλάση ServiceController του .NET Framework για να αποκτηθούν όλες οι υπηρεσίες και έπειτα με ένα “for loop” και ανάγνωση των κατάλληλων τιμών από το Registry δημιουργούνται αντικείμενα τύπου WindowsService τα οποία προστίθενται στη static λίστα. Παράλληλα λαμβάνει χώρα και ένα φιλτράρισμα ώστε να εμφανίζονται στο χρήστη μόνο όσες υπηρεσίες δεν θα προκαλέσουν σοβαρές επιπτώσεις αν τροποποιηθούν, δηλαδή αυτές που δεν αποτελούν κρίσιμο κομμάτι του συστήματος.

Η κλάση Disk Cleanup

Η κλάση DiskCleanup περιέχει τις μεθόδους που χρησιμοποιούνται για τον καθαρισμό του συστήματος από περιττά αρχεία. Οι μέθοδοι της πραγματοποιούν αναζήτηση αρχείων με φίλτρα και στα αποτελέσματα που επιστρέφονται καλούνται οι μέθοδοι διαγραφής DeleteFile και DeleteDir.

Επίσης περιέχει δύο μεθόδους για την εκκαθάριση των profiles των Mozilla Firefox και Google Chrome αντίστοιχα.

Η κλάση Package

Όπως και η WindowsService έτσι και η κλάση Package αποτελεί την αναπαράσταση της οντότητας ενός εγκατεστημένου προγράμματος. Εκτός από τα προφανή πεδία όπως PackageName, Publisher, Size που αντιπροσωπεύουν το όνομα, τον εκδότη και το μέγεθος ενός εγκατεστημένου προγράμματος, η κλάση

Package περιλαμβάνει και πεδία που περιέχουν την εντολή αντικατάστασης του προγράμματος καθώς και πεδία που επισημαίνουν αν είναι ασφαλής η κατάργηση του εν λόγω προγράμματος. Η κλάση Package υλοποιεί επίσης το interface IComparer που καθιστά δυνατή τη σύγκριση δύο αντικειμένων της κλάσης Package. Χάρη σε αυτό η λίστα των εγκατεστημένων προγραμμάτων που εντοπίζονται στο σύστημα μπορεί να ταξινομηθεί. Η ταξινόμηση γίνεται με αλφαβητική σειρά. Το όνομα της κλάσης είναι εμπνευσμένο από τις Linux διανομές που ο χρήστης εγκαθιστά πακέτα για να αποκτήσει νέα προγράμματα. Το αρχείο Program.cs υπάρχει ήδη σε κάθε C# πρόγραμμα οπότε η λέξη Package ήταν μια καλή εναλλακτική που περιγράφει αποτελεσματικά την οντότητα του εγκατεστημένου προγράμματος.

Η κλάση PackageManager

Η κλάση PackageManager αποτελεί μία από τις πιο σημαντικές κλάσεις στον AdvancedSystemManager για αυτό θα μελετηθεί λίγο περισσότερο η λειτουργία της σε βάθος. Η κλάση περιέχει δύο static λίστες τις InstalledProgramsList και startupProgramsList. Η δεύτερη περιέχει τα προγράμματα που εκτελούνται αυτόματα στην εκκίνηση ενώ η πρώτη με την οποία και θα ασχοληθούμε παρακάτω περιέχει τη λίστα των εγκατεστημένων προγραμμάτων που βρίσκονται στο σύστημα. Η λίστα αυτή γεμίζει καλώντας τη μέθοδο **GetAllProgramsList()** μια wrapper μέθοδο γύρω από άλλες μεθόδους που εντοπίζουν το εγκατεστημένο λογισμικό του υπολογιστή.

Στη συνέχεια καλούνται οι μέθοδοι **RemoveSystemComponents** και **RemoveUpdates** που αφαιρούν από τη λίστα περιττές εγγραφές που δεν εμφανίζονται στον πίνακα προσθαφαίρεσης προγραμμάτων των Windows. Οι εγγραφές αυτές αποτελούνται από ενημερώσεις των Windows και τμήματα των Windows που προέρχονται από ενημερώσεις ή από runtime πακέτα (για παράδειγμα Visual C++ Redistributable). Στη συνέχεια μέσω της μεθόδου **SortPackages** γίνεται η ταξινόμηση της λίστας με αλφαβητική σειρά. Τέλος κάθε πρόγραμμα της λίστας περνάει από κάποια φίλτρα προκειμένου να διαπιστωθεί αν μπορεί να αφαιρεθεί με ασφάλεια μέσα από την μέθοδο **IsSafeToRemove**.

Η κλάση πήρε το όνομα της εμπνευσμένη πάλι από τις Linux διανομές όπου ο package manager είναι το πρόγραμμα-οντότητα που εγκαθιστά/αντικαθιστά κάθε πρόγραμμα(πακέτο) στο σύστημα.

Σε αυτό το μοτίβο η κλάση PackageManager περιέχει μεθόδους με σκοπό τελικά την εγκατάσταση ή απεγκατάσταση προγραμμάτων στο σύστημα. Οι μέθοδοι

αυτές αν και δεν διαφέρουν σε πολλά σημεία στον τρόπο υλοποίησής τους εκτελούν τελικά διαφορετικές λειτουργίες.

Πιο συγκεκριμμένα έχουμε τις `MSI_Install(String filePath)`, `EXE_Install(String inEXEpath)` , `EXE_Uninstall(String uniEXEpath, String arg)` και `UninstallNoArgs(String args)`.

Όλες κάνουν χρήση της κλάσης `Process` του `.NET` .

Η `UninstallNoArgs` εκτελεί μια γραμμή εντολών των Windows (command prompt – cmd) σε ένα κρυφό παράθυρο από το χρήστη και σε αυτή την γραμμή εντολών εκτελείται η παράμετρος `args` που αποτελείται από το εκτελέσιμο αρχείο του `uninstaller` του επιλεγμένου προγράμματος και τυχόν παραμέτρους. Η `UninstallNoArgs` εκτελεί τόσο `.msi` όσο και `.exe` απεγκαταστάσεις.

Η `EXE_Uninstall` δέχεται σαν παραμέτρους το εκτελέσιμο αρχείο απεγκατάστασης και τις παραμέτρους σε ξεχωριστή παράμετρο. Χρησιμοποιώντας την κλάση `Process` αυτή τη φορά χωρίς όμως μια ενδιάμεση γραμμή εντολών αλλά εκτελώντας απευθείας το εκτελέσιμο απεγκατάστασης.

Η `EXE_Install` πραγματοποιεί κάτι αντίστοιχο με την `EXE_Uninstall` αλλά για διαδικασίες εγκατάστασης των αρχείων που βρίσκονται στο φάκελο `apps_deploy`.

Αντίστοιχα η `MSI_Install` πραγματοποιεί εγκαταστάσεις σαν την `EXE_Install` αλλά για `.msi` εκτελέσιμα αρχεία εγκατάστασης.

Η χρήση της σωστής μεθόδου απεγκατάστασης καθώς και ο ορισμός των σωστών σιωπηλών παραμέτρων γίνεται από τη μέθοδο **CheckUninstallationMethod** για κάθε πακέτο/πρόγραμμα που ήταν επιλεγμένο όταν πατηθεί το κουμπί **Optimize**.

Μέσα από την πολύπλοκη λογική του κώδικα μπορούμε να διακρίνουμε τις παρακάτω περιπτώσεις ενός **Uninstall String**, δηλαδή μιας συμβολοσειράς που περιλαμβάνει την εντολή απεγκατάστασης για το τρέχον επιλεγμένο πρόγραμμα προς απεγκατάσταση.

1) Το `Uninstall String` ξεκινάει με τη “φράση” `MsiExec.exe` , άρα έχουμε να κάνουμε με ένα πρόγραμμα που εγκαταστάθηκε μέσω ενός αρχείου `.msi` . Μέσω κατάλληλων μετασχηματισμών δίνουμε στην εντολή την εξής μορφή : `"MsiExec.exe /X" + UninstallString + " /qn"`; όπου `/X` η λειτουργία απεγκατάστασης και `/qn` η σιωπηλή/στο background εκτέλεση αυτής. Αν δεν έχουμε `Uninstall String` για `.msi` τότε πάμε στις επόμενες περιπτώσεις

2)`Uninstall String` που είναι `enquoted` δηλαδή περιέχεται μέσα σε εισαγωγικά (“ “):

2.a) Το String τελειώνει με τη “φράση” “.exe” άρα πρόκειται απλά για το path του εκτελέσιμου απεγκατάσης . Ο AdvancedSystemManager το καλεί με τις κατάλληλες σιωπηλές παραμέτρους.

3) Uninstall String που ξεκινά με (“) άνοιγμα εισαγωγικών) αλλά δεν τελειώνει με (“) κλείσιμο αυτών . Αυτό μαρτυρά ένα Uninstall String όπου η διαδρομή του εκτελέσιμου βρίσκεται μέσα σε εισαγωγικά(για αποφυγή κενών) ενώ οι παράμετροι του ακολουθούν χωρίς εισαγωγικά. Ο AdvancedSystemManager απλά εκτελεί το συγκεκριμένο Uninstall String.

4) Uninstall String που δεν ξεκινά και δεν τελειώνει με (“). Εδώ ο AdvancedSystemManager απλά φροντίζει να περάσει εισαγωγικά γύρω από τη διαδρομή του εκτελέσιμου και το εκτελεί όπως στην περίπτωση 2.

5)Uninstall String που δεν ξεκινά με (“) αλλά τελειώνει με (“). Αυτή είναι μια ειδική περίπτωση που συναντάται σε λίγα προγράμματα. Εδώ ο AdvancedSystemManager αφαιρεί όλα τα εισαγωγικά που υπάρχουν στο Uninstall String, στη συνέχεια ξαναπερνά εισαγωγικά μόνο γύρω από τη διαδρομή του εκτελέσιμου απεγκατάστασης ενώ προσθέτει τις σιωπηλές παραμέτρους και το εκτελεί.

Η λογική αυτή μπορεί να μελετηθεί αναλυτικότερα στον κώδικα της μεθόδου **CheckUninstallationMethod** της κλάσης PackageManager.

Η κλάση RegistryParser

Η κλάση RegistryParser αποτελεί ίσως την σημαντικότερη κλάση του AdvancedSystemManager καθώς πάρα πολλές λειτουργίες του προγράμματος υλοποιούνται εδώ. Όπως έχει ήδη αναφερθεί ο AdvancedSystemManager κάνει συχνή και μεγάλη χρήση του Registry για διάφορους λόγους εύρεσης πληροφοριών (εντοπισμός εγκατεστημένων προγραμμάτων, υπηρεσιών, έκδοσης Windows) αλλά ταυτόχρονα προσπαθεί να μην αλλοιώνει αυτές τις πληροφορίες όπως τα άλλα προγράμματα του εμπορίου παρά μόνο σε ελάχιστες περιπτώσεις που αυτό κρίνεται απαραίτητο(διαχείριση των startup items και services, visual effects, volume caches).

Οτιδήποτε λοιπόν έχει να κάνει με το μητρώο των Windows έχει να κάνει με την κλάση RegistryParser. Όλες οι άλλες κλάσεις για να επιτύχουν το σκοπό και τη λειτουργία τους χρησιμοποιούν λίγο πολύ μία μέθοδο της κλάσης RegistryParser.

Παρακάτω θα αναφερθούμε συνοπτικά στις πιο σημαντικές από αυτές,τον σκοπό που εξυπηρετούν καθώς και με ποιον τρόπο καλούνται από την εφαρμογή/χρήστη.

Πριν προχωρήσει περισσότερο ο αναγνώστης καλό θα ήταν να έχει γνώσεις της δομής του μητρώου των Windows διαβάζοντας το αντίστοιχο τμήμα από το κεφάλαιο 1.

Το .NET Framework προσφέρει κλάσεις για απευθείας πρόσβαση στο μητρώο. Η κλάση που θα χρησιμοποιήσουμε είναι η RegistryKey .

Μέσω της μεθόδου OpenSubKey που δέχεται σαν όρισμα τη διαδρομή ενός κλειδιού στο μητρώο σαν συμβολοσειρά μπορούμε να διαβάσουμε ή και να γράψουμε σε ένα κλειδί στο μητρώο.

Ανάλογα τη συμβολοσειρά που περνάμε σαν όρισμα διαβάζεται και το αντίστοιχο κλειδί.

Ενδεικτικά κάποιες διαδρομές κλειδιών είναι:

HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall

HKLM\Software\Wow6432Node\MicrosoftWindows\CurrentVersion\Uninstall(συναντάται σε 64bit λειτουργικά μόνο)

HKCU\Software\Microsoft\Windows\CurrentVersion\Uninstall

όπου ας θυμίσουμε ότι:

HKLM = HKEY_LOCAL_MACHINE

HKCU = HKEY_CURRENT_USER

Εδώ να θυμίσουμε επίσης ότι οι τιμές/διαδρομές των κλειδιών του μητρώου δεν είναι case-sensitive, δηλαδή το κλειδί

“HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall” και το

“HKLM\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\UNINSTALL”

αντιπροσωπεύουν το ίδιο κλειδί στο μητρώο.

Η μέθοδος WindowsVersion

Η μέθοδος WindowsVersion επιστρέφει σε μια συμβολοσειρά (String) την έκδοση του λειτουργικού συστήματος που εκτελείται στο σύστημα . Αυτό το κάνει διαβάζοντας το κλειδί HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion . Η WindowsVersion καλείται από την κλάση SystemInfo κατά τη διάρκεια εντοπισμού των πληροφοριών του συστήματος

Η μέθοδος GetVolumeCaches

Αν και οι μέθοδοι καθαρισμού του δίσκου από περιττά αρχεία βρίσκονται στη μέθοδο DiskCleanup, οι τοποθεσίες που καλείται να ψάξει η DiskCleanup βρίσκονται αποθηκευμένες στο μητρώο και πιο συγκεκριμένα στο κλειδί “HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches”. Το ίδιο μάλιστα κλειδί χρησιμοποιεί το ενσωματωμένο εργαλείο καθαρισμού των δίσκων των Windows (cleanmgr.exe) για να εντοπίσει τα απαραίτητα αρχεία προς διαγραφή. Η GetVolumeCaches καλεί για κάθε τιμή του κλειδιού τις μεθόδους της DiskCleanup όταν ο χρήστης επιλέξει την επιλογή Disk Cleanup (ή -cleandisk στο command line περιβάλλον αντίστοιχα).

Η μέθοδος ApplyVisualEffects

Η μέθοδος ApplyVisualEffects αποτελεί μία από τις ελάχιστες εξαιρέσεις που ο AdvancedSystemManager τροποποιεί/γράφει τιμές στο μητρώο. Η ApplyVisualEffects είναι η μέθοδος που απενεργοποιεί ορισμένα οπτικά εφέ των Windows για να παραμείνουν όμως απενεργοποιημένα και μετά την επανεκκίνηση του συστήματος. Η διαφορά αυτής με την ApplySettings της κλάσης VisualEffects είναι στο ότι η δεύτερη απενεργοποιεί τα οπτικά εφέ μόνο για την τωρινή συνεδρία του χρήστη χωρίς να τα εφαρμόζει σε κάθε εκκίνηση. Ουσιαστικά είναι απαραίτητη η ύπαρξη και των δυο μεθόδων γιατί η Microsoft δεν παρέχει μέσω του .NET ή από unsafe code (DllImport) τη δυνατότητα επαναεφαρμογής των ρυθμίσεων του DWM(Desktop Window Manager) που είναι υπεύθυνος για τα οπτικά εφέ που συμβαίνουν στο Windows Shell.[12]

Η μέθοδος GetStartupPrograms

Όπως μαρτυρά και το όνομά της, η συγκεκριμένη μέθοδος διαβάσει τις τιμές των κλειδιών αυτόματης εκκίνησης (HKLM\Software\Microsoft\Windows\CurrentVersion\Run (και HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run για τα 64bit συστήματα) και προσθέτει στη λίστα StartupProgramsList της κλάσης PackageManager.

Η μέθοδος GetDisabledStartupPrograms

Αντίστοιχα η μέθοδος αυτή προσθέτει στη λίστα τα ανενεργά πλέον κατά την εκκίνηση προγράμματα . Αυτό επιτυγχάνεται διαβάζοντας το κλειδί “HKLM\

Software\Microsoft\Shared Tools\MSConfig\startupreg” . Έτσι εμφανίζεται η πλήρης λίστα των προγραμμάτων εκκίνησης στο χρήστη στην καρτέλα Startup Manager.

Οι μέθοδοι EnableKeyStartup και DisableKeyStartup

Οι παραπάνω μέθοδοι καλούνται όταν ο χρήστης αλλάξει την κατάσταση εκκίνησης σε κάποιο πρόγραμμα στην καρτέλα Startup Manager. Η λειτουργία τους είναι η μεταφορά από τα ενεργά στα ανενεργά στοιχεία εκκίνησης και το αντίστροφο. Να υπενθυμίσουμε ότι αυτές οι μέθοδοι καλούνται άμεσα, δηλαδή με το που ο χρήστης “τσεκάρει” ένα πρόγραμμα στην καρτέλα Startup Manager.

Η μέθοδος GetPrograms

Η παραπάνω μέθοδος χρησιμοποιείται για την συμπλήρωση της λίστας InstalledProgramsList που βρίσκεται στην κλάση PackageManager. Πιο συγκεκριμένα στο κλειδί “HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall” υπάρχουν δεκάδες υποκλειδιά από τα οποία κάθε ένα αντιπροσωπεύει ένα εγκατεστημένο πρόγραμμα. Διαβάζοντας κατάλληλα τις τιμές, για κάθε υποκλειδί καλείται ο δομητής της κλάσης Package και δημιουργείται ένα αντικείμενο τύπου Package το οποίο εισέρχεται στην λίστα InstalledProgramsList.

Η μέθοδος GetWin64Programs

Η μέθοδος GetWin64Programs καλείται μόνο σε περίπτωση που το τρέχων λειτουργικό σύστημα είναι 64bit. Παρόμοια με την GetPrograms συμπληρώνει τη λίστα με προγράμματα που έχουν αποθηκευμένες τις πληροφορίες τους στο reflected-key (καθρεφτισμένο κλειδί) “HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall”

Η μέθοδος GetCurrentUserPrograms

Τέλος η GetCurrentUserPrograms ενημερώνει τη λίστα με τυχόν εγγραφές που βρίσκονται στο κλειδί “HKCU\Software\Microsoft\Windows\CurrentVersion\Uninstall” . Τέτοιες εγγραφές προγράμματα είναι συνήθως τα λεγόμενα portable ή προγράμματα που εμφανίζουν την επιλογή κατά τη διάρκεια της εγκατάστασης να εγκατασταθούν μόνο για τον τωρινό χρήστη (current user).

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό είδαμε μερικές από τις πιο σημαντικές κλάσεις και πως αυτές χρησιμοποιούνται στο πρόγραμμά μας. Αναλύσαμε σε βάθος τις κλάσεις και τις μεθόδους ανάγνωσης και εγγραφής στο μητρώο καθώς και τον τρόπο διαχείρισης των εγκατεστημένων προγραμμάτων από την εφαρμογή μας. Στο επόμενο κεφάλαιο θα κάνουμε μια σύγκριση του AdvancedSystemManager με άλλες αντίστοιχες εφαρμογές του εμπορίου που προσφέρουν παρόμοιες δυνατότητες και λειτουργίες και θα αναφέρουμε υποψήφιες βελτιώσεις για την εφαρμογή.

ΚΕΦΑΛΑΙΟ 5

ΣΥΓΚΡΙΣΗ ΤΟΥ ADVANCED SYSTEM MANAGER ΜΕ ΑΛΛΕΣ ΛΥΣΕΙΣ ΤΟΥ ΕΜΠΟΡΙΟΥ – ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΕΠΕΚΤΑΣΗ

ΕΙΣΑΓΩΓΗ

Τώρα που προηγήθηκε μια ανάλυση της δομής και του τρόπου λειτουργίας του AdvancedSystemManager μπορούμε να προχωρήσουμε σε μια σύγκριση της εφαρμογής μας σε σχέση με τις υπόλοιπες διαθέσιμες του εμπορίου καθώς και να προτείνουμε μελλοντικά κομμάτια επέκτασης και βελτίωσης του AdvancedSystemManager.

Σύγκριση του Advanced System Manager με άλλα διαθέσιμα optimization εργαλεία

Ίσως η πιο σημαντική διαφορά του Advanced System Manager σε σχέση με τα άλλα optimization εργαλεία του εμπορίου είναι ότι αποτελεί ελεύθερο και ανοιχτού κώδικα λογισμικό. Αυτό σημαίνει ότι ο πηγαίος κώδικας του προγράμματος είναι διαθέσιμος για ανάγνωση από τον οποιονδήποτε για να διαπιστώσει και ο ίδιος πως υλοποιούνται όλες οι λειτουργίες που προσφέρει αλλά και να βεβαιωθεί ότι δεν υπάρχει οτιδήποτε κακόβουλο στον κώδικα. Κάτι τέτοιο φυσικά δεν μπορεί να συμβεί στα προγράμματα κλειστού κώδικα. Ακόμα το γεγονός ότι ο AdvancedSystemManager αποτελεί ελεύθερο λογισμικό, δίνει το δικαίωμα στον καθένα να τροποποιήσει τον κώδικα του και να τον προσαρμόσει στις ανάγκες του. Κοινώς αποκτάται και το χαρακτηριστικό της επεκτασιμότητας αλλά και μελλοντικής βελτίωσης από τρίτα άτομα. Άλλες διαφορές ακολουθούν παρακάτω:

- Ο Advanced System Manager δεν αποκρύπτει από το χρήστη τα πραγματικά, ενσωματωμένα εργαλεία των Windows που χρησιμοποιούνται ή προσωμειώνεται η λειτουργία τους μέσα στο πρόγραμμα. Αντίθετα κάνει αισθητή την παρουσία τους στην καρτέλα Tools.
- Το πρόγραμμα δεν “τρομοκρατεί” το χρήστη με ανύπαρκτα προβλήματα στο Windows Registry και ψεύτικες διορθώσεις. Συγκεκριμένα δεν εφαρμόζει και δεν ασχολείται με τέτοιες διορθώσεις καθαρισμού του μητρώου. Όπως είναι γραμμένο και στους επίσημους οδηγούς συντήρησης της Microsoft η χρήση των λεγόμενων “Registry Cleaners” δεν υποστηρίζεται επίσημα και δεν προσφέρει παραπάνω επιδόσεις. Αντίθετα είναι πιθανό να οδηγήσει σε σφάλματα.

- Ο AdvancedSystemManager αν και διαβάζει μεγάλο πλήθος πληροφοριών από το μητρώο των Windows σπάνια κάνει χρήση της δυνατότητας εγγραφής αλλά και όπου αυτό γίνεται, είναι σε ελεγχόμενα τμήματα που αποκλείεται να οδηγήσουν σε αστάθεια, ζημιά ή αδυναμία εκκίνησης του συστήματος. Συγκεκριμένα τα μόνα τμήματα που γράφονται ποτέ είναι τα στοιχεία εκκίνησης, ο τύπος εκκίνησης μια υπηρεσίας και οι ρυθμίσεις οπτικών εφέ των Windows. Έτσι επιτυγχάνεται η εγγυημένη σωστή λειτουργία του συστήματος καθώς δεν υπάρχει αυθαίρετη διαγραφή/τροποποίηση κλειδιών που ενδεχομένως να χρειάζονται από το σύστημα.
- Ο AdvancedSystemManager μπορεί να εκτελέσει τις περισσότερες από τις λειτουργίες του και σε περιβάλλον κονσόλας, γεγονός που τον καθιστά ιδιαίτερο χρήσιμο σε περιπτώσεις αυτοματοποίησης και μαζικής εγκατάστασης σε πολλούς υπολογιστές.
- Το μέγεθος του εκτελέσιμου αρχείου είναι μικρότερο από 100KB , πράγμα που επιτρέπει τη λήψη του AdvancedSystemManager ακόμα και από τις πιο αργές συνδέσεις. Την εποχή της ευρυζωνικότητας αυτό δεν αποτελεί και ιδιαίτερα μεγάλο πλεονέκτημα καθώς και οι αντίστοιχες λύσεις του εμπορίου κυμαίνονται μεταξύ των 10 και 40 MB, ποσό σίγουρα όχι πολύ μεγάλο. Σίγουρα όμως καθίσταται η λήψη του αρκετά ευκολότερη σε χώρες που χρησιμοποιούν ακόμα dial-up συνδέσεις.
- Ο AdvancedSystemManager παρέχει όλες τις βασικές Hardware πληροφορίες του συστήματος που εκτελείται που μπορεί να χρειάζεται ένας χρήστης. Τα άλλα εργαλεία του εμπορίου δεν διαθέτουν αυτή τη δυνατότητα και έτσι είναι απαραίτητη είτε η γνώση εντοπισμού των πληροφοριών μέσα από τα Windows (ανάγκη αναζήτησης σε τρεις τοποθεσίες για εξαγωγή όλων των πληροφοριών) είτε η χρήση ενός ακόμα εξωτερικού εργαλείου (για παράδειγμα το δημοφιλές CPU-Z)
- Ο AdvancedSystemManager παρέχει και δυνατότητα σιωπηλής μαζικής εγκατάστασης προγραμμάτων ενώ οι ήδη υπάρχουσες λύσεις προσφέρουν μόνο δυνατότητα απεγκατάστασης (και όχι πάντα μαζικής)

Προτάσεις για μελλοντική επέκταση της εφαρμογής

Μερικές πιθανές βελτιώσεις και νέες λειτουργίες που θα μπορούσαν να προστεθούν στον AdvancedSystemManager ακολουθούν παρακάτω:

- Λήψη γνωστών open source προγραμμάτων χρησιμοποιώντας μια υπηρεσία web και εγκατάσταση αυτών στην περίπτωση που δεν βρεθούν προγράμματα προς εγκατάσταση στο φάκελο apps_deploy αλλά ο χρήστης χρησιμοποίησε την επιλογή “Unattended Installation” κατά τη διάρκεια του optimization.
- Λήψη Runtime βιβλιοθηκών (για παράδειγμα MSVC++ Redistributable) για επίλυση σφαλμάτων εκτέλεσης.
- Καθαρισμός από όλες τις επαναλήψεις μέσα στον κώδικα. Κυρίως στην κλάση PackageManager και RegistryParser υπάρχουν αρκετά κομμάτια κώδικα που επαναλαμβάνονται.
- Δημιουργία test cases για ευκολότερο έλεγχο των αποτελεσμάτων των λειτουργιών του προγράμματος.
- Επέκταση του command line interface της εφαρμογής ώστε να υποστηρίζει όλες τις λειτουργίες του προγράμματος.
- Δημιουργία διεπαφής χρήστη με δυνατότητα για πολλαπλές μεταφράσεις (multi-language GUI)
- Απεγκλωβισμός του νήματος κατά τη διάρκεια μαζικής (απ)εγκατάστασης για τα προγράμματα που εμφανίζουν υποχρεωτικά διαλόγους λόγω των custom installers τους.
- Δημιουργία περισσότερων ρυθμίσεων εφαρμογής φίλτρων για τον τρόπο επιλογής λογισμικού προς απεγκατάσταση

ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό παρουσιάσαμε μερικά από τα πλεονεκτήματα του AdvancedSystemManager έναντι των άλλων διαθέσιμων εμπορικών εφαρμογών και μερικές πιθανές επεκτάσεις για την εφαρμογή. Ακολουθούν τα συμπεράσματα της πτυχιακής εργασίας.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο σκοπός αυτής της πτυχιακής εργασίας ήταν να προσφέρει στον αρχάριο χρήστη ένα ισχυρό εργαλείο για την εύκολη διατήρηση και συντήρηση του Windows συστήματός του. Σύμφωνα με την γνώμη του συγγραφέα αυτός ο στόχος επιτεύχθηκε σε μεγάλο βαθμό. Σίγουρα υπάρχουν περιθώρια βελτίωσης όπως συμβαίνει με κάθε κομμάτι λογισμικού τα οποία θα μπορούσαν να αποτελούν το θέμα και κάποιας άλλης πτυχιακής εργασίας. Ακολουθεί η βιβλιογραφία και ο οδηγός χρήσης του προγράμματος AdvancedSystemManager.

ΑΝΑΦΟΡΕΣ

- 1 <https://technet.microsoft.com/en-us/library/cc768129.aspx>
- 2 <https://technet.microsoft.com/library/cc767881.aspx>
- 3 <https://support.microsoft.com/en-us/help/256986/windows-registry-information-for-advanced-users>
- 4 <https://support.microsoft.com/en-us/help/314866/a-definition-of-the-run-keys-in-the-windows-xp-registry>
- 5 [https://msdn.microsoft.com/en-us/library/aa394582\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx)
- 6 <https://msdn.microsoft.com/en-us/library/aa389273.aspx>
- 7 <https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/walkthrough-running-an-operation-in-the-background>
- 8 <https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/how-to-run-an-operation-in-the-background>
- 9 <https://support.microsoft.com/en-us/help/2563254/microsoft-support-policy-for-the-use-of-registry-cleaning-utilities>
- 10 <https://blogs.technet.microsoft.com/markrussinovich/2005/10/02/registry-junk-a-windows-fact-of-life/>
- 11 [https://technet.microsoft.com/en-us/library/cc748940\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc748940(v=ws.10).aspx)
- 12 <https://technet.microsoft.com/en-us/library/cc957204.aspx>

ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

Παρακάτω ακολουθεί ένας σύντομος οδηγός χρήσης του Advanced System Manager τόσο για το γραφικό περιβάλλον του όσο και τις εντολές που υποστηρίζει σε περιβάλλον κονσόλας.

Έχοντας κατεβάσει το AdvancedSystemManager.zip επιλέγουμε το αρχείο και κάνουμε αποσυμπίεση με το πρόγραμμα της επιλογής μας (ή ακόμα και με τον ενσωματωμένο zip unpacker των Windows).

Στο φάκελο που δημιουργείται έχουμε τα παρακάτω αρχεία και φακέλους:

AdvancedSystemManager.exe : το βασικό πρόγραμμα που εκτελούμε με διπλό κλικ

remove.list : πρόκειται για ένα αρχείο κειμένου που περιλαμβάνει μια λίστα γνωστών “κακόφημων” προγραμμάτων που προτείνονται για αφαίρεση. Εφόσον είναι ένα απλό αρχείο κειμένου ο χρήστης μπορεί να προσθέσει ή να αφαιρέσει προγράμματα από τη λίστα, διευκολύνοντας έτσι στην αυτοματοποίηση της διαδικασίας απεγκατάστασης προγραμμάτων.

Φάκελος apps_deploy : Εδώ μπορεί ο χρήστης να τοποθετήσει εκτελέσιμα αρχεία εγκατάστασης εφαρμογών (setup executable files). Υποστηριζόμενες επεκτάσεις είναι οι .exe και .msi . Αρχεία άλλων μορφών αγνοούνται. Αν ο χρήστης έχει επιλέξει την λειτουργία Unattended Installation κατά τη διάρκεια του Optimization (ανάλυση της λειτουργίας Optimization ακολουθεί παρακάτω) ή τρέξει τον AdvancedSystemManager από τη γραμμή εντολών με την παράμετρο “-install” (χωρίς τα εισαγωγικά) , τότε ο AdvancedSystemManager θα ψάξει για αρχεία εγκατάστασης προγραμμάτων στο φάκελο apps_deploy και θα προσπαθήσει να τα εγκαταστήσει “σιωπηλά” στο σύστημα, δηλαδή χωρίς να εμφανιστούν οι τυπικοί διάλογοι “Next/Next/Next” στον χρήστη. Αυτό βέβαια δεν επιτρέπει την τροποποίηση/επιλογή από το χρήστη για παράδειγμα για το φάκελο εγκατάστασης αλλά προσφέρει σίγουρα μια μεγάλη ευκολία σε περιπτώσεις που χρειάζεται η εγκατάσταση πολλών προγραμμάτων σε πολλούς υπολογιστές ταυτόχρονα ή δεν υπάρχει αρκετός χρόνος για την πραγματοποίηση της κάθε εγκατάστασης ξεχωριστά από το χρήστη.

Κατά τη διάρκεια της εκτέλεσης του AdvancedSystemManager ή και μετά τη λήξη της διάρκειας αυτής υπάρχει η πιθανότητα να εμφανιστούν διάφορα αρχεία με επέκταση .log . Πάλι πρόκειται για απλά αρχεία κειμένου που περιέχουν μηνύματα αποσφαλμάτωσης(debugging) και δεν πρέπει η ύπαρξη τους να προβληματίζει τον χρήστη. Μπορούν να διαγραφούν με ασφάλεια και καθαρίζονται αυτόματα από το

προηγούμενο περιεχόμενό τους σε κάθε εκκίνηση του προγράμματος. Περιέχουν πληροφορίες για το αν προέκυψε κάποιο σφάλμα ή κάποια λειτουργία του AdvancedSystemManager δεν εκτελέστηκε σωστά.

Advanced System Manager – Γραφικό Περιβάλλον

Το γραφικό περιβάλλον του AdvancedSystemManager αποτελείται από μία κύρια οθόνη καθορισμένων διαστάσεων (δεν μπορεί να μεγιστοποιηθεί ή να αλλάξουν οι διαστάσεις του κυρίως παραθύρου) και έναν άξονα με καρτέλες(tabs) στην κορυφή ακριβώς όπως και οι καρτέλες σε έναν περιηγητή ιστού (web browser). Σε αντίθεση βέβαια με έναν περιηγητή ιστού οι καρτέλες δεν μπορούν να “κλείσουν”. Ο χρήστης εξακολουθεί όμως να μπορεί να εναλλάσει ανάμεσα σε αυτές επιλέγοντας κάθε φορά με το ποντίκι την καρτέλα που επιθυμεί να μεταβεί.

Η καρτέλα System Info

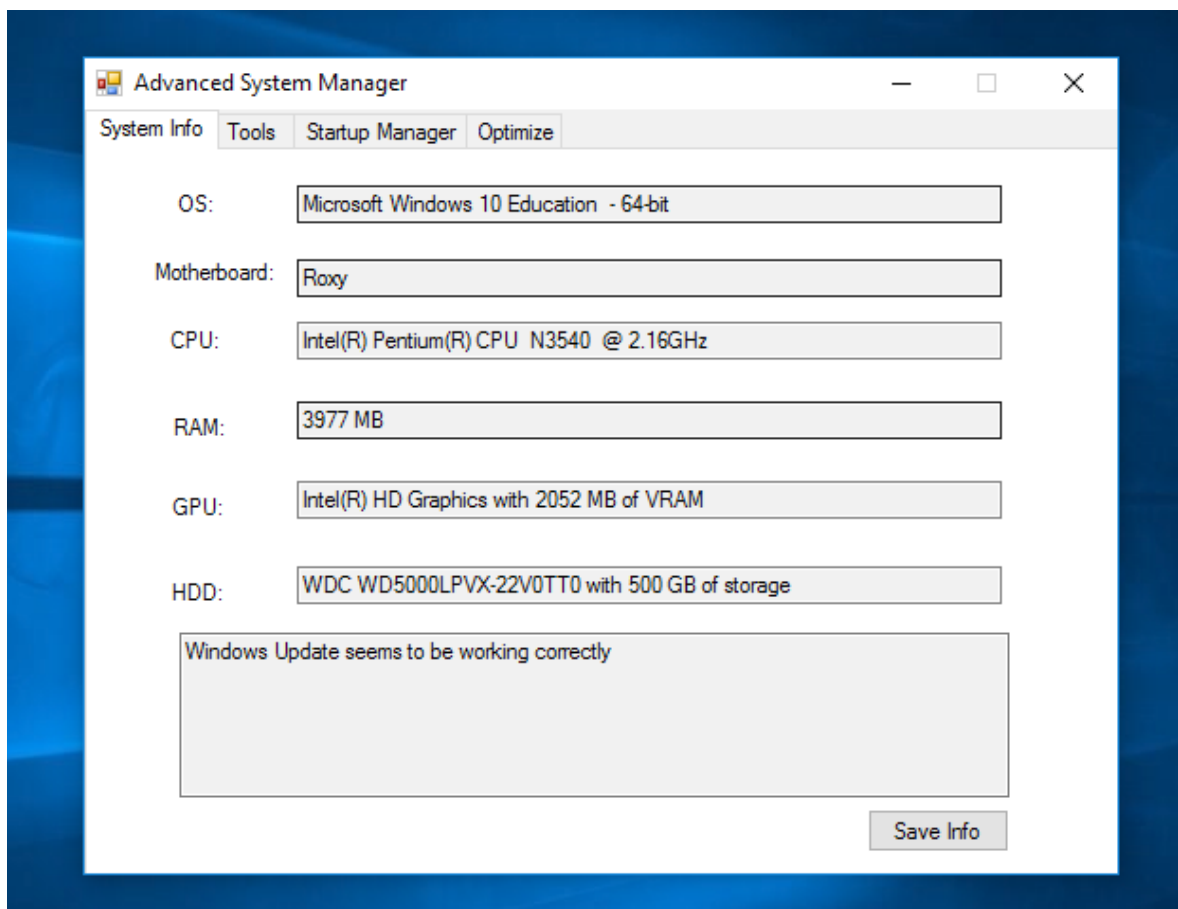
Στην καρτέλα System Info ο χρήστης μπορεί να δει πληροφορίες για το hardware και την έκδοση των Windows που υπάρχουν στον υπολογιστή του. Η καρτέλα System Info παρέχει όλες τις βασικές πληροφορίες που χρειάζεται να γνωρίζει ο χρήστης για τον υπολογιστή του σε περίπτωση ανάγκης επικοινωνίας με κάποιον τεχνικό. Οι πληροφορίες αυτές είναι:

- η έκδοση των Windows και η αρχιτεκτονική τους (32 ή 64bit)
- Το μοντέλο της μητρικής κάρτας
- Το μοντέλο του επεξεργαστή του συστήματος
- Η συνολική μνήμη RAM του συστήματος εκφρασμένη σε MegaBytes
- Το μοντέλο της κάρτας γραφικών και η ποσότητα της μνήμης της σε MegaBytes
- Το μοντέλο του σκληρού δίσκου και η μέγιστη χωρητικότητα του εκφρασμένη σε GigaBytes

Στο κάτω μέρος της καρτέλας υπάρχει ένα πλαίσιο με πληροφορίες για την λειτουργία του Windows Update. Πιο συγκεκριμένα στα συστήματα με Windows 7 και Windows 8/8.1 γίνεται έλεγχος για την ύπαρξη συγκεκριμένων ενημερώσεων στο σύστημα που είναι υπεύθυνες για την χρήση του νέου Windows Update API. Αν δεν βρεθούν στο σύστημα ο χρήστης ενημερώνεται με κατάλληλο μήνυμα στο συγκεκριμένο πλαίσιο. Στα συστήματα με Windows 10 που ήδη γίνεται χρήση του

νέου API όπως και σε αυτά με XP και Vista που πλέον δεν υποστηρίζονται από την Microsoft το συγκεκριμένο πλαίσιο δεν προσφέρει κάποια αξιόπιστη πληροφορία στο χρήστη.

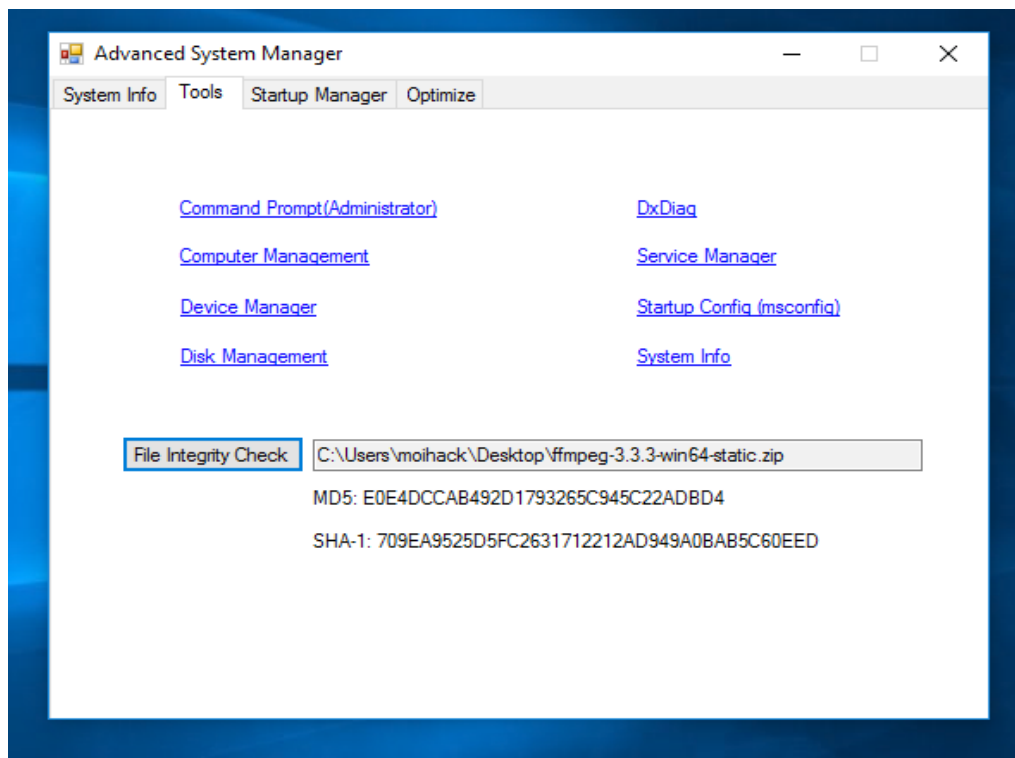
Τέλος το κουμπί Save Info κάτω δεξιά αποθηκεύει τις πληροφορίες του συστήματος στο αρχείο SystemInfo.txt στον ίδιο φάκελο που βρίσκεται και το ίδιο το εκτελέσιμο του AdvancedSystemManager. Αν το αρχείο δεν υπάρχει ήδη δημιουργείται αυτόματα, ενώ αν υπάρχει απλά επαναγράφεται το περιεχόμενό του με τις πληροφορίες του τρέχοντος συστήματος. Η καρτέλα System Info φαίνεται στην παρακάτω εικόνα:



Εικόνα 19: Η καρτέλα SystemInfo

Η καρτέλα Tools

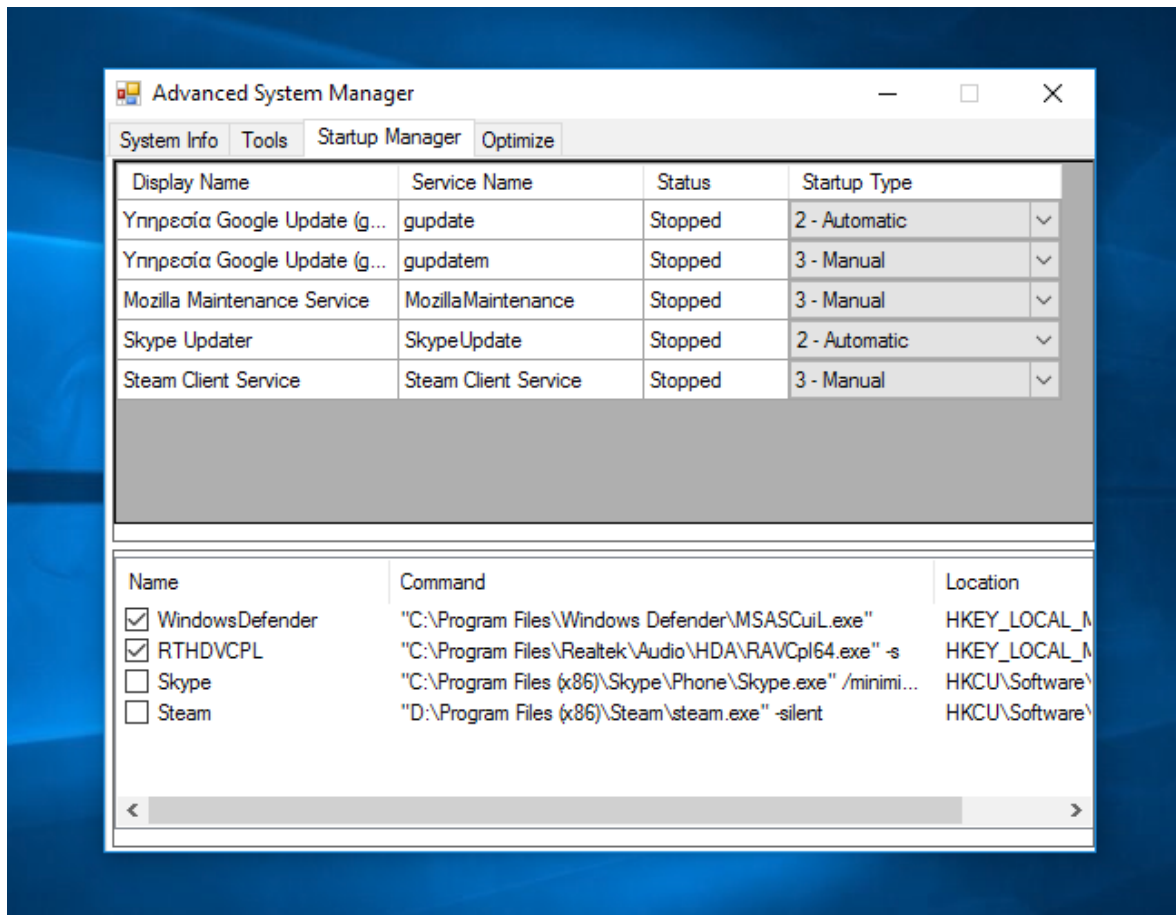
Η καρτέλα Tools περιλαμβάνει συντομεύσεις για γνωστά εργαλεία διαχείρισης των Windows, την ύπαρξη των οποίων συχνά αγνοεί ο αρχάριος χρήστης. Για τον πιο ειδικό χρήστη αποτελεί ένα πίνακα ελέγχου για γρήγορα πρόσβαση σε αυτά τα εργαλεία. Στο κάτω μέρος της καρτέλας υπάρχει το κουμπί **Integrity Check**. Πατώντας το ο χρήστης καλείται να επιλέξει ένα από τα αρχεία που έχει αποθηκευμένα στον υπολογιστή του και ο AdvancedSystemManager θα υπολογίσει τα αθροίσματα ελέγχου MD5 και SHA-1 για το επιλεγμένο αρχείο. Δυνατότητα υπολογισμού αθροισμάτων ελέγχου δεν υπάρχει στα Windows συστήματα με κάποιο ενσωματωμένο στο λειτουργικό σύστημα εργαλείο όπως συμβαίνει για παράδειγμα στις Linux διανομές (md5sum) και έτσι ο χρήστης αναγκάζεται να καταφύγει σε ξεχωριστά εργαλεία (όπως το fciv της Microsoft – διαθέσιμο μόνο σε περιβάλλον κονσόλας) για να πραγματοποιήσει τέτοιες λειτουργίες ελέγχου ακεραιότητας αρχείων. Το περιβάλλον της καρτέλας Tools φαίνεται στην παρακάτω εικόνα:



Εικόνα 20: Η καρτέλα Tools

Η καρτέλα Startup Manager

Στην καρτέλα Startup Manager ο χρήστης μπορεί να διαχειριστεί τις υπηρεσίες (services) και τα προγράμματα του υπολογιστή που ξεκινούν αυτόματα με την εκκίνηση των Windows. Πιο συγκεκριμένα αυτή η καρτέλα χωρίζεται σε δύο τμήματα ένα πάνω και ένα κάτω όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 21: Η καρτέλα Startup Manager

Στο επάνω τμήμα της διεπαφής βρίσκεται η λίστα με τις υπηρεσίες που εκτελούνται(ή εκτελούνταν κάποτε) στην εκκίνηση των Windows. Στην πρώτη στήλη (Display Name) βρίσκεται το "φιλικό" προς το χρήστη όνομα της υπηρεσίας, ένα όνομα που ο χρήστης μπορεί να καταλάβει συνοπτικά την λειτουργία της υπηρεσίας. Στη δεύτερη στήλη (Service Name) βρίσκεται το όνομα της υπηρεσίας όπως είναι γνωστό στο λειτουργικό σύστημα. Το όνομα αυτό δεν μεταφράζεται ακόμα και αν τρέχουμε μια μεταφρασμένη έκδοση των Windows σε αντίθεση με το πρώτο το οποίο μπορεί να μεταφράζεται ανάλογα τη γλώσσα του συστήματός μας. Στην στήλη Status(Κατάσταση) μπορούμε να δούμε την κατάσταση της κάθε

υπηρεσίας η οποία μπορεί να είναι είτε Running(σε εκτέλεση) είτε Stopped(σταματημένη). Τέλος στην τέταρτη στήλη Startup Type ο χρήστης μπορεί να δει ή και να τροποποιήσει το τύπο εκκίνησης κάθε υπηρεσίας. Ο τύπος εκκίνησης μπορεί να είναι **Automatic**, **Manual** και **Disabled**. Η ερμηνεία του κάθε τύπου φαίνεται παρακάτω:

Automatic : αυτόματη εκκίνηση της επιλεγμένης υπηρεσίας μαζί με τα Windows

Manual : χειροκίνητη εκκίνηση της επιλεγμένης υπηρεσίας είτε από τον ίδιο το χρήστη είτε από μία άλλη υπηρεσία που εξαρτάται από αυτή(τη manual υπηρεσία).

Disabled: απενεργοποιεί τελείως την υπηρεσία. Τυχόν υπηρεσίες που εξαρτώνται από αυτήν θα οδηγήσουν σε σφάλμα. Για να ενεργοποιήσει ο χρήστης την συγκεκριμένη υπηρεσία ξανά μπορεί είτε να αναθέσει μια άλλη τιμή από τις διαθέσιμες μέσα από τον AdvancedSystemManager είτε από το εργαλείο διαχείρισης υπηρεσιών των Windows(services.msc) το οποίο μπορεί εύκολα να βρεί και από την καρτέλα Tools.

Επειδή η διαχείριση των υπηρεσιών είναι μια προχωρημένη διαδικασία και η απενεργοποίηση των λάθος υπηρεσιών μπορεί να οδηγήσει τα Windows σε αποτυχία εκκίνησης του συστήματος ή σε σφάλματα σε άλλες λειτουργίες, ο AdvancedSystemManager έχοντας σαν κύριο στόχο τον αρχάριο χρήστη αποκρύπτει (εσκεμμένα) από τη λίστα υπηρεσιών, όλες τις υπηρεσίες συστήματος(των Windows, hardware drivers κτλ) ώστε να αποφευχθούν τυχόν ανεπιθύμητες παρενέργειες από λάθος επιλογές του χρήστη. Για να έχει πρόσβαση ο χρήστης σε όλες τις υπηρεσίες του συστήματος με οτιδήποτε αυτό συνεπάγεται θα πρέπει να καταφύγει στη χρήση των εργαλείων mscsonfig ή services.msc (προσβάσιμα και τα δύο από την καρτέλα Tools).

Στο δεύτερο (κάτω) μέρος της καρτέλας Startup Manager υπάρχει η λίστα των στοιχείων/προγραμμάτων που εκκινούν αυτόματα ή εκκινούσαν παλαιότερα κατά την διαδικασία εκκίνησης των Windows.

Στην στήλη Name φαίνεται το όνομα του προγράμματος που εκκινεί/εκκινούσε αυτόματα.

Παρόλα αυτά το όνομα δεν είναι πάντα επεξηγηματικό όπως συμβαίνει με τις υπηρεσίες και συχνά χρησιμοποιείται κ η δεύτερη στήλη (Command) για τον προσδιορισμό της λειτουργίας των στοιχείων. Η στήλη Command δείχνει την πλήρη διαδρομή του εκτελέσιμου καθώς και τυχόν παραμέτρους που δέχεται κατά την εκτέλεση του. Η στήλη Command μπορεί να οδηγήσει πολλές φορές στον προσδιορισμό της λειτουργίας ενός στοιχείου εκκίνησης. Για παράδειγμα στην

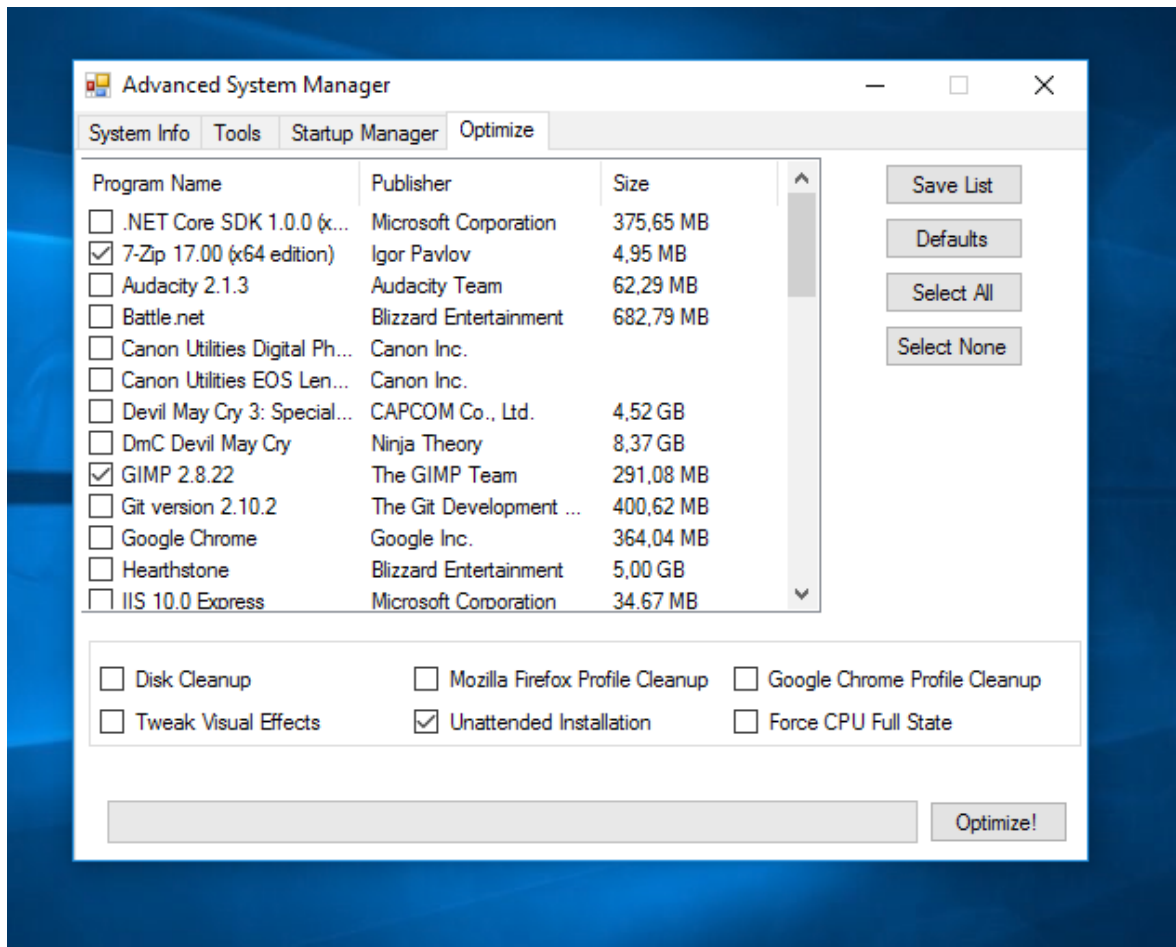
παρακάτω εικόνα το δεύτερο στοιχείο φέρει ένα όνομα(RTHDVCPL) που δεν σημαίνει κάτι για το χρήστη. Παρατηρώντας όμως που βρίσκεται το εκτελέσιμο του στη στήλη Command ο χρήστης μπορεί να καταλάβει ότι πρόκειται για το Control Panel μιας Realtek κάρτας ήχου. Τέλος η στήλη Location δείχνει στο χρήστη την τοποθεσία του κλειδιού(key) στο μητρώο των Windows σε περίπτωση που ο χρήστης θέλει χειροκίνητα να εντοπίσει το κλειδί και να το απενεργοποιήσει μέσω του μητρώου ή να αναζητήσει περισσότερες πληροφορίες για αυτό στο διαδίκτυο.

Να σημειωθεί ότι κάθε αλλαγή που πραγματοποιείται στην καρτέλα Startup Manager εφαρμόζεται άμεσα στο σύστημα αλλά για να γίνουν αντιληπτές οι αλλαγές από το χρήστη είναι απαραίτητη η επανεκκίνηση του συστήματος. Η καρτέλα Startup Manager προσομοιώνει τις καρτέλες Services και Startup του εργαλείου msconfig (προσβάσιμο από την καρτέλα Tools).

Η καρτέλα Optimize

Η καρτέλα Optimize είναι ίσως η σημαντικότερη του AdvancedSystemManager καθώς εδώ συμβαίνουν οι πιο πολύπλοκες λειτουργίες όπως η δυνατότητα μαζικής εγκατάστασης/απεγκατάστασης προγραμμάτων καθώς και η εφαρμογή επιλεγμένων βελτιώσεων στο σύστημα που αυξάνουν τις επιδόσεις.

Η καρτέλα Optimize χωρίζεται και αυτή σε δύο μέρη (ακριβώς σαν την Startup Manager, πάνω και κάτω μέρος αντίστοιχα) όπως φαίνεται στην εικόνα:



Εικόνα 22: Η καρτέλα Optimize

Στο πάνω μέρος εμφανίζεται μια πανομοιότυπη λίστα, ακριβώς ίδια με αυτήν από τον πίνακα προσθαφαίρεσης προγραμμάτων των Windows. Ο χρήστης μπορεί να επιλέξει τα προγράμματα που επιθυμεί προς αφαίρεση ενεργοποιώντας το checkbox δίπλα στο όνομα του κάθε προγράμματος της λίστας. Από μόνος του ο AdvancedSystemManager ήδη έχει επιλέξει κάποια προγράμματα για αφαίρεση βασιζόμενος σε κάποια φίλτρα ενσωματωμένα στον κώδικα καθώς και στις εγγραφές στο αρχείο "remove.list" (αν υπάρχει). Στα δεξιά της λίστας υπάρχουν 4 κουμπιά.

Το "Save List" θα αποθηκεύσει την λίστα των εγκατεστημένων προγραμμάτων στο αρχείο programs.txt στον ίδιο φάκελο που βρίσκεται το εκτελέσιμο του AdvancedSystemManager. Αν το αρχείο δεν υπάρχει θα δημιουργηθεί, ενώ αν υπάρχει το περιεχόμενό του θα αντικατασταθεί από τη λίστα.

Το κουμπί “Defaults” θα επιλέξει μόνο τα προγράμματα που είχε επιλέξει ο AdvancedSystemManager αρχικά. Χρήσιμο σε περίπτωση που ο χρήστης επιλέξει/αποεπιλέξει κάποια προγράμματα εκτός των ήδη αυτόματα επιλεγμένων και τελικά το μετανιώσει.

Τα κουμπιά “Select All” και “Select None” όπως μαρτυρά και η ονομασία τους επιλέγουν και αποεπιλέγουν όλα τα προγράμματα αντίστοιχα. Κανένα πρόγραμμα δεν θα αφαιρεθεί προς το παρόν μέχρις ότου ο χρήστης πατήσει το κουμπί **Optimize** που βρίσκεται στο κάτω δεξί τμήμα του παραθύρου.

Περνώντας στο δεύτερο(κάτω) τμήμα της καρτέλας Optimize έχουμε ένα πλήθος από διαφορετικά checkboxes. Κάθε επιλογή προσφέρει μέσα στο πρόγραμμα περιγραφή της λειτουργίας της αν φέρει ο χρήστης το ποντίκι από πάνω της, όμως θα αναλύσουμε και εδώ την κάθε επιλογή ξεχωριστά.

Όπως και με τη λίστα προγραμμάτων επάνω, καμία επιλογή δεν είναι μη-αναστρέψιμη και η εφαρμογή της στο σύστημα θα αρχίσει μόνο αν πατήσει ο χρήστης το κουμπί Optimize.

Ας δούμε όμως την λειτουργία των επιλογών:

- **Disk Cleanup** : Αναλαμβάνει να εντοπίσει στο σκληρό δίσκο αρχεία που χρησιμοποιούνται για σκοπούς αποσφαλμάτωσης και καταγραφής καθώς και προσωρινά αρχεία παλαιών εγκαταστάσεων και τα διαγράφει. Δεν διαγράφονται αρχεία του χρήστη καθώς τα αρχεία που διαγράφονται βρίσκονται σε όλα τα συστήματα σε φακέλους που κανένας χρήστης δεν χρησιμοποιεί για αποθήκευση των αρχείων του.

- **Mozilla Firefox Profile Cleanup** : Ο πολύ δημοφιλής browser της Mozilla χρησιμοποιεί έναν ξεχωριστό, κρυφό φάκελο για να αποθηκεύει τις ρυθμίσεις/προτιμήσεις του χρήστη(αγαπημένα,κωδικοί πρόσβασης,ιστορικό,επεκτάσεις κτλ). Αν ο χρήστης εγκαθιστά λογισμικό από αναξιόπιστες πηγές ή αγνοεί τις προτροπές για επιπλέον ανεπιθύμητο λογισμικό κατά τη διάρκεια μιας εγκατάστασης , είναι πιθανό ο Firefox να αποκτήσει περιττά toolbars, αλλαγμένη αρχική σελίδα ή ακόμα και ενοχλητικά pop-ups. Αν και αυτά τα προβλήματα συνήθως αντιμετωπίζονται εύκολα απλά μέσα από τις ρυθμίσεις του Firefox υπάρχουν περιπτώσεις κακόβουλου λογισμικού(Conduit toolbar) που επιμένουν και επανεγγράφουν τις ρυθμίσεις του browser. Σε τέτοιες περιπτώσεις μια ολική διαγραφή του profile/φακέλου των ρυθμίσεων μπορεί να βοηθήσει. Φυσικά τυχόν ρυθμίσεις(αγαπημένα,κωδικοί,ιστορικό) θα χαθούν , οπότε είναι καλό να γίνει πρώτα συγχρονισμός τους με κάποια υπηρεσία online αποθήκευσης.

- **Google Chrome Profile Cleanup** : Όμοια με τον Firefox μπορεί και ο Chrome να χρειαστεί μια ολική επαναφορά ρυθμίσεων για να απαλλαγεί από κάποιο “επίμονο” κακόβουλο λογισμικό.

- **Tweak Visual Effects** : Απενεργοποιεί κάποιες από τις ρυθμίσεις των “οπτικών εφέ” των Windows με σκοπό την ταχύτερη λειτουργία του γραφικού περιβάλλοντος και την αύξηση των επιδόσεων. Ενδεικτικά κάποια από τα εφέ που απενεργοποιούνται είναι οι σκιές γύρω από τα παράθυρα και το ποντίκι, η διαφάνεια των παραθύρων, τα animations της ελαχιστοποίησης και της μεγιστοποίησης και άλλα. Δεν απενεργοποιείται το Windows Aero ούτε το Compositing και έτσι ο χρήστης δύσκολα μπορεί να αντιληφθεί μια οπτική διαφορά στον τομέα της εμφάνισης, παρά μόνο στην ταχύτητα. Για να εφαρμοστούν πλήρως όλες οι αλλαγές είναι απαραίτητη μια επανεκκίνηση του συστήματος.

- **Unattended Installation** : Όταν είναι τσεκαρισμένο ο AdvancedSystemManager θα ψάξει στο φάκελο apps_deploy για τυχόν εκτελέσιμα αρχεία εγκατάστασης και θα προσπαθήσει να τα εγκαταστήσει σιωπηλά(silent installation) αμέσως μόλις τελειώσει με την απεγκατάσταση όσων τυχόν προγραμμάτων επιλέχθηκαν.

- **Force CPU Full State** : Όλοι οι σύγχρονοι επεξεργαστές έρχονται ενσωματωμένοι με μηχανισμούς εξοικονόμησης ενέργειας. Οι μηχανισμοί αυτοί πραγματοποιούν την αυξομείωση και προσαρμογή της συχνότητας του επεξεργαστή ανάλογα με το φόρτο του και τη διάρκεια του έχουν ως σκοπό την εξοικονόμηση ενέργειας. Πολλές φορές όμως οι επιδόσεις περιορίζονται από αυτές τις αυξομειώσεις με αποτέλεσμα να δυσχεραίνει η εμπειρία του χρήστη. Με αυτή την επιλογή ο επεξεργαστής του συστήματος θα λειτουργεί πάντα στο μέγιστο της συχνότητας του, καταναλώνοντας επιπλέον ενέργεια και παράγοντας επιπλέον θερμότητα, αυξάνοντας όμως άμεσα τις επιδόσεις σε πολλές εργασίες.

AdvancedSystemManager – Command Line Περιβάλλον

Ο AdvancedSystemManager μπορεί να εκτελεστεί και σε περιβάλλον κονσόλας πράγμα χρήσιμο για περιπτώσεις αυτοματοποίησης και διαχείρισης πολλαπλών συστημάτων καθώς έτσι καθίσταται εύκολη η χρήση του μέσα σε scripts.

Ο AdvancedSystemManager δεν υποστηρίζει όλες του τις λειτουργίες σε περιβάλλον κονσόλας ούτε και επιτρέπει την παραμετροποίηση αυτών όπως γίνεται στο γραφικό του περιβάλλον. Οι παράμετροι που υποστηρίζονται είναι:

-sysinfo (αποθηκεύει στο αρχείο info.txt τις πληροφορίες του συστήματος (Windows, μοντέλο επεξεργαστή, ποσότητα RAM κτλ)

-getprograms (αποθηκεύει στο αρχείο info.txt την λίστα των εγκατεστημένων προγραμμάτων του τρέχοντος υπολογιστή)

-effects (εφαρμόζει την κατάργηση ορισμένων οπτικών εφέ των Windows για αύξηση των επιδόσεων – απαιτείται επανεκκίνηση)

-remove (απεγκαθιστά όσα προγράμματα επιλέγονται αυτόματα από τον AdvancedSystemManager καθώς και αυτά που βρίσκονται δηλωμένα στο αρχείο remove.list. Μπορεί να αφαιρεθεί και κάποιο πρόγραμμα που επιθυμεί να κρατήσει ο χρήστης καθώς δεν δίνεται η δυνατότητα να αποεπιλέξει κάποιο αν το επιθυμεί.

-install (εγκαθιστά όσα προγράμματα έχουν εκτελέσιμα αρχεία εγκατάστασης στο φάκελο apps_deploy)

-clean (καθαρίζει το δίσκο από προσωρινά αρχεία παλαιότερων εγκαταστάσεων καθώς και από αρχείο ελέγχου και αποσφαλμάτωσης)

Κάθε άλλη παράμετρος αντιμετωπίζεται ως άγνωστη από τον AdvancedSystemManager και αυτό θα έχει ως αποτέλεσμα να τερματίζεται η εφαρμογή άμεσα.

Σημείωση: Όλες οι παράμετροι χρειάζονται την παύλα(-) να προηγείται και είναι case sensitive δηλαδή πρέπει να γράφονται με μικρά γράμματα ακριβώς όπως αναγράφονται σε αυτό τον οδηγό. Διαφορετικά θα αντιμετωπισθούν ως άγνωστες παράμετροι.