



**Ανώτατο Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Κρήτης**

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΗ ΕΙΚΟΝΙΚΗ ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ



Σπουδαστής: Μπαζάκας Νικόλαος

Αριθμός Μητρώου: 813

Επιβλέπων Καθηγητής: Μαλάμος Αθανάσιος

Νοέμβριος 2009

Περιεχόμενα

Πρόλογος.....	6
Αντικείμενο της πτυχιακής εργασίας	6
Κεφάλαιο 1.....	7
1.1 Internet.....	7
1.1.1 Γενικά.....	7
1.1.2 Η ιστορία του Διαδικτύου	7
1.1.3 Το Internet Σήμερα.....	9
1.1.4 Διαδικτυακές υπηρεσίες και πρωτόκολλα.....	9
1.2 Πρωτόκολλο Μεταφοράς Αρχείων (File Transfer Protocol - FTP)	10
1.2.1 Γενικά.....	10
1.2.2 Ιστορική Αναδρομή.....	10
1.2.3 Τρόπος λειτουργίας	11
1.2.4 Χρήση.....	12
1.2.5 Anonymous FTP.....	12
1.2.6 Ασφάλεια.....	13
Κεφάλαιο 2.....	15
2.1 Γενικά.....	15
2.2 BestSynch.....	15
2.3 FTP Synchronization Software.....	17
2.4 FTP Synchronizer Professional	19
2.5 Συγκριση της πτυχιακής	24
Κεφάλαιο 3.....	26
3.1.Γενικά.....	26
3.2 Microsoft Visual Studio	26
3.2.1 Χαρακτηριστικά Γνωρίσματα	27
3.2.2 Συμπεριλαμβανόμενα Προϊόντα.....	31
3.3 Microsoft Visual Basic	32
3.3.1 Γενικά	32
3.3.2 Χαρακτηριστικά της Γλώσσας	32

3.4 Microsoft .NET Framework	34
3.4.1 Γενικά.....	34
3.4.2 Base Class Library.....	34
3.4.3 Χρόνος Εκτέλεσης Κοινής Γλώσσας	34
3.4.4 Κύρια Χαρακτηριστικά Σχεδιασμού	35
3.4.5 .NET Framework 3.5.....	36
Κεφάλαιο 4.....	38
4.1 Γενικά.....	38
4.2 Βασικές Λειτουργίες της Εφαρμογής.....	38
4.3 Συστατικά της εφαρμογής	39
4.3.1 Η Κλάση CommonClasses	39
4.3.2 LoginForm.....	44
4.3.3 DownloadForm.....	46
4.3.4 WidgetForm.....	49
Κεφάλαιο 5.....	58
Εγχειρίδιο Χρήσης	58
Βιβλιογραφία	63

**“Give a person a fish and you feed him for a day;
teach that person to use the Internet and he won't bother you for weeks.”**

Πρόλογος

Αντικείμενο της πτυχιακής εργασίας

Σκοπός της παρούσας πτυχιακής είναι η δημιουργία μιας διαδικτυακής εφαρμογής η οποία θα επιτρέπει στον χρήστη την απομακρυσμένη διαχείριση εικονικά των αρχείων που υπάρχουν σε ένα διακομιστή FTP χρησιμοποιώντας το πρωτόκολλο μεταφοράς αρχείων (File Transfer Protocol - FTP).

Η εφαρμογή κατά την εκκίνηση της δημιουργεί ένα φάκελο στην επιφάνεια εργασίας του τοπικού υπολογιστή, στον οποίο κατεβάζει όλα τα περιεχόμενα του φακέλου του απομακρυσμένου διακομιστή FTP. Ο χρήστης μπορεί να διαχειριστεί τα περιεχόμενα του φακέλου που δημιούργησε η εφαρμογή σαν να διαχειριζόταν ένα οποιοδήποτε αρχείο ή φάκελο του προσωπικού του υπολογιστή.

Κατά τον τερματισμό της, ή όποτε το επιλέξει ο χρήστης, η εφαρμογή εντοπίζει όλες τις τροποποιήσεις και τις προσθήκες που έγιναν στα περιεχόμενα του φακέλου που δημιούργησε στην εκκίνηση της, και πραγματοποιεί τις απαραίτητες αλλαγές στον φάκελο του απομακρυσμένου διακομιστή FTP.

Με αυτό τον τρόπο καταφέρνουμε να έχουμε ανα πάσα στιγμή διαθέσιμα για επεξεργασία ή ανάγνωση τα αρχεία που έχουμε επιλέξει σε οποιοδήποτε υπολογιστή κι αν βρισκόμαστε.

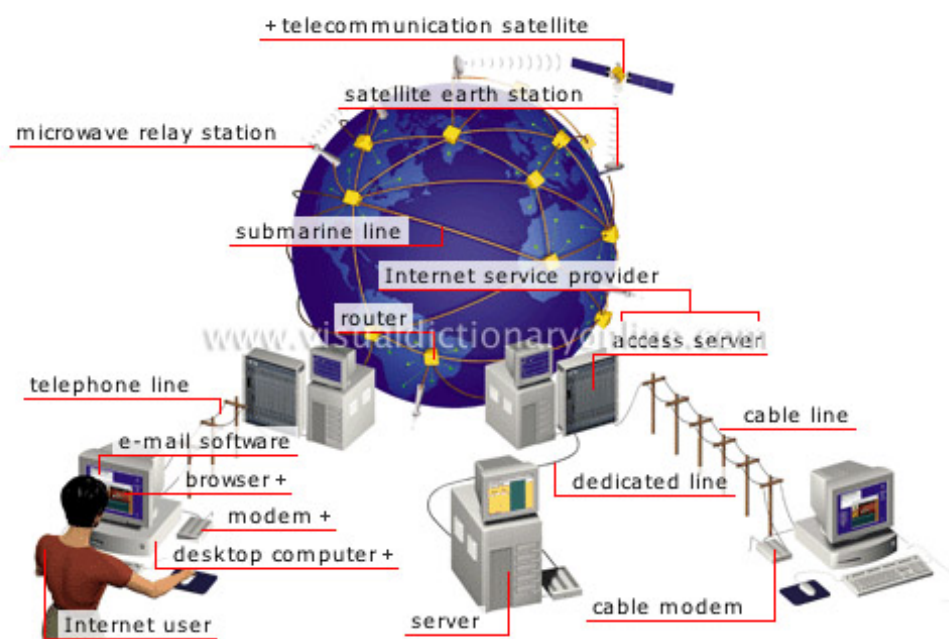
Η εφαρμογή αυτή αναπτύχθηκε στο περιβάλλον του Microsoft Visual Studio 2008 χρησιμοποιώντας ως γλώσσα προγραμματισμού την Visual Basic. Η επιλογή του περιβάλλοντος εργασίας και της γλώσσας προγραμματισμού έγινε με γνώμονα την ταχύτητα και την ευελιξία που μας προσφέρει η Microsoft μέσα από τη χρήση του .Net Framework.

Κεφάλαιο 1

1.1 Internet

1.1.1 Γενικά

Το **Διαδίκτυο** ή **Ίντερνετ** (Internet) είναι ένα επικοινωνιακό δίκτυο ηλεκτρονικών υπολογιστών, που επιτρέπει την ανταλλαγή δεδομένων μεταξύ οποιουδήποτε διασυνδεδεμένου υπολογιστή. Η τεχνολογία του είναι κυρίως βασισμένη στην διασύνδεση επιμέρους δικτύων ανά τον κόσμο και πολυάριθμα τεχνολογικά πρωτόκολλα, με κύριο το TCP/IP. Ο αντίστοιχος αγγλικός όρος *internet* προκύπτει από τη σύνθεση λέξεων *inter-network*. Στην πιο εξειδικευμένη και περισσότερο χρησιμοποιούμενη μορφή του, με τους όρους **Διαδίκτυο**, **Ίντερνέτ** ή **Ίντερνετ** (με κεφαλαίο το αρχικό γράμμα) περιγράφεται το παγκόσμιο πλέγμα διασυνδεδεμένων υπολογιστών και των υπηρεσιών και πληροφοριών που παρέχει στους χρήστες του. Το Διαδίκτυο χρησιμοποιεί μεταγωγή πακέτων (*packet switching*) και τη στοίβα πρωτοκόλλων TCP/IP.



Εικόνα 1.1 Τρόποι Λειτουργίας Του Διαδικτύου
(Πηγή: http://www.ac-nancy-metz.fr/enseign/anglais/henry/internet_1.jpg)

1.1.2 Η ιστορία του Διαδικτύου

Οι πρώτες απόπειρες για την δημιουργία ενός διαδικτύου ξεκίνησαν στις ΗΠΑ κατά την διάρκεια του ψυχρού πολέμου. Η Ρωσία είχε ήδη στείλει στο διάστημα τον δορυφόρο

Σπούτνικ 1 κάνοντας τους Αμερικανούς να φοβούνται όλο και περισσότερο για την ασφάλεια της χώρας τους. Θέλοντας λοιπόν να προστατευτούν από μια πιθανή πυρηνική επίθεση των Ρώσων δημιούργησαν την υπηρεσία προηγμένων αμυντικών ερευνών ARPA (Advanced Research Project Agency) γνωστή ως DARPA (Defense Advanced Research Projects Agency) στις μέρες μας. Αποστολή της συγκεκριμένης υπηρεσίας ήταν να βοηθήσει τις στρατιωτικές δυνάμεις των ΗΠΑ να αναπτυχθούν τεχνολογικά και να δημιουργηθεί ένα δίκτυο επικοινωνίας το οποίο θα μπορούσε να επιβιώσει σε μια ενδεχόμενη πυρηνική επίθεση.

Το αρχικό θεωρητικό υπόβαθρο δόθηκε από τον Τζ. Λικλάιντερ (J.C.R. Licklider) που ανέφερε σε συγγράμματά του το "γαλαξιακό δίκτυο". Η θεωρία αυτή υποστήριζε την ύπαρξη ενός δικτύου υπολογιστών που θα ήταν συνδεδεμένοι μεταξύ τους και θα μπορούσαν να ανταλλάσσουν γρήγορα πληροφορίες και προγράμματα. Το επόμενο θέμα που προέκυπτε ήταν ότι το δίκτυο αυτό θα έπρεπε να ήταν αποκεντρωμένο έτσι ώστε ακόμα κι αν κάποιος κόμβος του δεχόταν επίθεση να υπήρχε δίοδος επικοινωνίας για τους υπόλοιπους υπολογιστές. Τη λύση σε αυτό έδωσε ο Πολ Μπάραν (Paul Baran) με τον σχεδιασμό ενός καταναμημένου δικτύου επικοινωνίας που χρησιμοποιούσε την ψηφιακή τεχνολογία. Πολύ σημαντικό ρόλο έπαιξε και η θεωρία ανταλλαγής πακέτων του Λέοναρντ Κλάινροκ (Leonard Kleinrock), που υποστήριζε ότι πακέτα πληροφοριών που θα περιείχαν την προέλευση και τον προορισμό τους μπορούσαν να σταλούν από έναν υπολογιστή σε έναν άλλο.

Η δημιουργία του ARPANET

Στηριζόμενο λοιπόν σε αυτές τις τρεις θεωρίες δημιουργήθηκε το πρώτο είδος διαδικτύου γνωστό ως ARPANET. Εγκαταστάθηκε και λειτούργησε για πρώτη φορά το 1969 με 4 κόμβους μέσω των οποίων συνδέονται 4 μίνι υπολογιστές (mini computers 12k): του πανεπιστημίου της Καλιφόρνια στην Σάντα Μάρμπαρα, του πανεπιστημίου της Καλιφόρνια στο Λος Άντζελες, το SRI στο Στάνφορντ και το πανεπιστήμιο της Γιούτα. Η ταχύτητα του δικτύου έφθανε τα 50 kbps και έτσι επιτεύχθηκε η πρώτη *dial up* σύνδεση μέσω γραμμών τηλεφώνου. Μέχρι το 1972 οι συνδεδεμένοι στο ARPANET υπολογιστές έχουν φτάσει τους 23, οπότε και εφαρμόζεται για πρώτη φορά το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου(e-mail).

Παράλληλα δημιουργήθηκαν και άλλα δίκτυα, που χρησιμοποιούσαν διαφορετικές μεθόδους και τεχνικές(όπως το x.25 και το UUCP) τα οποία συνδέονταν με το ARPANET. Το πρωτόκολλο που χρησιμοποιούσε το ARPANET ήταν το NCP (Network Control Protocol), το οποίο, όμως, είχε το μειονέκτημα ότι λειτουργούσε μόνο με συγκεκριμένους τύπους υπολογιστών. Έτσι, δημιουργήθηκε η ανάγκη στις αρχές του 1970 για ένα πρωτόκολλο που θα ένωνε όλα τα δίκτυα που είχαν δημιουργηθεί μέχρι τότε. Το 1974 λοιπόν, δημοσιεύεται η μελέτη των Βιντ Σερφ (Vint Cerf) και Μπομπ Κάαν (Bob Kahn) από την οποία προέκυψε το πρωτόκολλο TCP (Transmission Control Protocol) που αργότερα το 1978 έγινε TCP/IP,

προστέθηκε δηλαδή το Internet Protocol (IP), και τελικά το 1983 έγινε το μοναδικό πρωτόκολλο που ακολουθούσε το ARPANET.

Η δημιουργία του πρώτου DNS (Domain Name System)

Το 1984 υλοποιείται το πρώτο DNS (Domain Name System) σύστημα στο οποίο καταγράφονται 1000 κεντρικοί κόμβοι και οι υπολογιστές του διαδικτύου πλέον αναγνωρίζονται από διευθύνσεις κωδικοποιημένων αριθμών. Ένα ακόμα σημαντικό βήμα στην ανάπτυξη του Διαδικτύου έκανε το Εθνικό Ίδρυμα Επιστημών (National Science Foundation, NSF) των ΗΠΑ, το οποίο δημιούργησε την πρώτη διαδικτυακή πανεπιστημιακή ραχοκοκκαλιά (backbone), το NSFNet, το 1986. Ακολούθησε η ενσωμάτωση άλλων σημαντικών δικτύων, όπως το Usenet, το Fidonet και το Bitnet.

Ο όρος Διαδίκτυο/Ίντερνετ ξεκίνησε να χρησιμοποιείται ευρέως την εποχή που συνδέθηκε το APRANET με το NSFNet και Ίντερνετ σήμαινε οποιοδήποτε δίκτυο χρησιμοποιούσε TCP/IP. Η μεγάλη άνθιση του Διαδικτύου όμως, ξεκίνησε με την εφαρμογή της υπηρεσίας του Παγκόσμιου Ιστού από τον Τιμ Μπέρνερς-Λι στο ερευνητικό ίδρυμα CERN το 1989, ο οποίος είναι, στην ουσία, η πλατφόρμα, η οποία κάνει εύκολη την πρόσβαση στο Ίντερνετ, ακόμα και στη μορφή που το ξέρουμε εμείς σήμερα. (Πηγή: <http://el.wikipedia.org/wiki/Διαδίκτυο>)

1.1.3 Το Internet Σήμερα

Σήμερα, το Internet έχει εισχωρήσει για τα καλά στη ζωή μας και αποτελεί πλέον μέρος της καθημερινότητας μας. Πιο συγκεκριμένα, σύμφωνα με την ιστοσελίδα Internet World Stats, το έτος 2009 σε πληθυσμό 6,767,805,208 ανθρώπων παγκοσμίως, καταγράφηκαν 1,668,870,408 χρήστες του Internet, αριθμός που αντιστοιχεί σε ένα ποσοστό της τάξεως του 24.7%, παρατηρώντας μια αύξηση κατά 362.3 % την τελευταία δεκαετία. Συγκεκριμένα στην Ευρώπη, σε συνολικό πληθυσμό 803,850,858 ανθρώπων, καταγράφηκαν 402,380,474 χρήστες του Internet, αριθμός που αντιστοιχεί σε ποσοστό 50,1%, ενώ παρατηρήθηκε μια αύξηση της τάξεως του 282.9% την τελευταία δεκαετία

1.1.4 Διαδικτυακές υπηρεσίες και πρωτόκολλα

Μερικά από τα πιο γνωστά διαδικτυακά πρωτόκολλα είναι τα TCP, IP, UDP, DNS, PPP, SLIP, ICMP, POP3, IMAP, SMTP, HTTP, HTTPS, SSH, Telnet, FTP, LDAP και το SSL. Μερικές από τις πιο γνωστές Διαδικτυακές υπηρεσίες που χρησιμοποιούν αυτά τα πρωτόκολλα είναι το ηλεκτρονικό ταχυδρομείο (*e-mail*), οι ομάδες συζητήσεων (*newsgroups*), η διαμοίραση αρχείων (*file sharing*), η ροή μέσων σε πραγματικό χρόνο (*streaming media*), η τηλεφωνία μέσω IP (*voice telephony – VoIP*), ο Παγκόσμιος Ιστός (*World Wide Web*) καθώς και η μεταφορά αρχείων (*file transfer*) με την οποία θα

ασχοληθούμε στην παρούσα εργασία.



Εικόνα 1.2 Μερικές Χρήσεις Του Internet

(Πηγή: <http://visual.merriam-webster.com/images/communications/office-automation/internet-uses.jpg>)

1.2 Πρωτόκολλο Μεταφοράς Αρχείων (File Transfer Protocol - FTP)

1.2.1 Γενικά

Το **Πρωτόκολλο Μεταφοράς Αρχείων** (File Transfer Protocol - FTP) είναι ένα ευρέως χρησιμοποιούμενο πρωτόκολλο σε δίκτυα τα οποία υποστηρίζουν το πρωτόκολλο TCP/IP (δίκτυα όπως internet ή intranet). Ο υπολογιστής που τρέχει εφαρμογή FTP client μόλις συνδεθεί με τον server μπορεί να εκτελέσει ένα πλήθος διεργασιών όπως ανέβασμα αρχείων στον server, κατέβασμα αρχείων από τον server, μετονομασία ή διαγραφή αρχείων από τον server κ.ο.κ. Το πρωτόκολλο είναι ένα ανοιχτό πρότυπο. Είναι δυνατό κάθε υπολογιστής που είναι συνδεδεμένος σε ένα δίκτυο, να διαχειρίζεται αρχεία σε ένα άλλο υπολογιστή του δικτύου, ακόμη και εάν ο δεύτερος διαθέτει διαφορετικό λειτουργικό σύστημα.

1.2.2 Ιστορική Αναδρομή

Οι πρώτοι κανόνες για ανάπτυξη μηχανισμών ανταλλαγής αρχείων απαντώνται το 1971, όταν αναπτύχθηκαν για χρήση στο M.I.T. Περιγράφονται στο RFC 114 και ακολουθούν σχόλια και συζήτηση πάνω στο RFC 141. Αξίζει να αναφερθεί το RFC 354, το οποίο αντικατέστησε τα ξεπερασμένα RFC 264 και RFC 265. Στο νέο RFC το FTP οριζόταν ως πρωτόκολλο ανταλλαγής αρχείων ανάμεσα σε hosts του δικτύου ARPANET. Πρωταρχικοί

στόχοι του πρωτοκόλλου ήταν η εύκολη και αξιόπιστη μεταφορά αρχείων και η εύκολη αξιοποίηση των δυνατοτήτων για απομακρυσμένη αποθήκευση αρχείων.

Τελικά, δημοσιεύτηκε ένα επίσημο έγγραφο για το FTP στο RFC 454. Ακολούθησαν και άλλα επίσημα RFCs αλλά, αν και μέχρι τον Ιούλιο του 1973 είχαν γίνει σημαντικές αλλαγές από τις τελευταίες εκδόσεις του, η δομή του παρέμενε ίδια. Πολλές υλοποιήσεις (implementations) του πρωτοκόλλου βασίζονταν σε παλαιότερες μη ενημερωμένες εκδόσεις.

Όλες οι παραπάνω προσπάθειες για δημιουργία ενός πρωτοκόλλου μεταφοράς αρχείων σε συνδυασμό με την μετάβαση από το NCP στο TCP, οδήγησαν στην επανεγγραφή του FTP στο RFC 765 το 1980. Σήμερα το κυριότερο RFC για το FTP είναι το RFC 959 ένα κείμενο εξήντα σελίδων.

Σε μεταγενέστερα RFCs σχετικά με το FTP, είτε περιγράφηκαν αναλυτικότερα ορισμένα σημεία του RFC 959 (όπως στο RFC 1635), είτε έγιναν προτάσεις σχετικές με την ασφάλειά του (όπως στο RFC 1579 και στο RFC 2228).

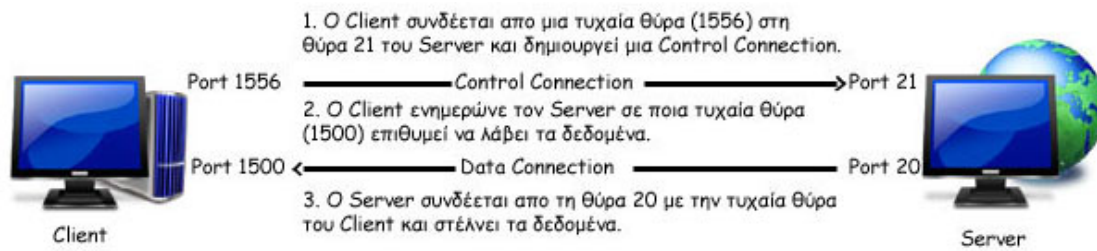
1.2.3 Τρόπος λειτουργίας

Αρχικά ο FTP server ανοίγει την θύρα (port) 21 περιμένοντας έναν FTP client να συνδεθεί. Στη συνέχεια ο client ξεκινά μια νέα σύνδεση από μια τυχαία θύρα προς την θύρα 21 του server. Μόλις γίνει η σύνδεση παραμένει ανοιχτή για όλη τη διάρκεια της συνόδου FTP. Η συγκεκριμένη σύνδεση ονομάζεται σύνδεση ελέγχου (control connection).

Έπεται η δημιουργία της σύνδεσης δεδομένων (data connection), της σύνδεσης με την οποία μεταφέρονται τα δεδομένα. Υπάρχουν δύο τρόποι για να δημιουργηθεί, με χρήση της ενεργητικής λειτουργίας (active mode) ή με χρήση της παθητικής λειτουργίας (passive mode).

1.2.3.1 Active mode

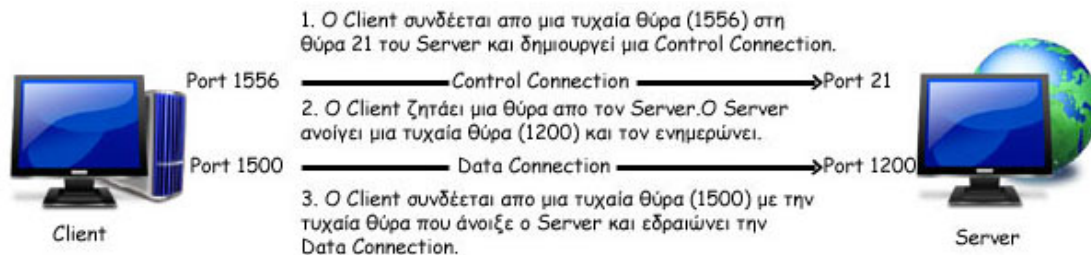
Στην ενεργητική λειτουργία (active mode) ο FTP client διαλέγει μια τυχαία θύρα στην οποία δέχεται τα δεδομένα της σύνδεσης. Ο client στέλνει τον αριθμό της θύρας, στην οποία επιθυμεί να "ακούει" (listen) για εισερχόμενες συνδέσεις. Ο FTP server δημιουργεί μια σύνδεση από την θύρα 20 στην ανοιχτή θύρα του client για τη μεταφορά των δεδομένων. Οποιαδήποτε πληροφορία ζητήσει ο client, ανταλλάσσεται με βάση αυτή τη σύνδεση, που βασίζεται στο TCP. Όταν η μεταφορά ολοκληρωθεί ο server κλείνει τη σύνδεση αποστέλλοντας ένα πακέτο FIN, όπως σε κάθε σύνδεση βασισμένη στο TCP. Κάθε φορά που ο client ζητάει δεδομένα, δημιουργείται κατά παρόμοιο τρόπο μια σύνδεση δεδομένων και η διαδικασία επαναλαμβάνεται.



Εικόνα 1.3 Ο τρόπος σύνδεσης στην ενεργητική λειτουργία

1.2.3.2 Passive mode

Στην παθητική λειτουργία (passive mode) ο client ζητά από τον server να διαλέξει μια τυχαία θύρα, στην οποία θα "ακούει" (listen) για την σύνδεση δεδομένων (data connection). Ο server ενημερώνει τον client για την θύρα την οποία έχει διαλέξει και ο client συνδέεται σε αυτή για τη μεταφορά των δεδομένων. Η μεταφορά ολοκληρώνεται όπως και στην ενεργητική λειτουργία (active mode), αφού η σύνδεση δεδομένων βασίζεται στο TCP.



Εικόνα 1.4 Ο τρόπος σύνδεσης στην παθητική λειτουργία

1.2.4 Χρήση

Το FTP είναι ένα πρωτόκολλο πελάτη-εξυπηρετητή (Client - Server) 8-bit, ικανό να χειρίζεται οποιοδήποτε τύπο αρχείου χωρίς περαιτέρω. Ωστόσο το FTP έχει εξαιρετικά υψηλή λανθάνουσα κατάσταση (latency). Αυτό σημαίνει ότι ο χρόνος μεταξύ του αιτήματος και της διαδικασίας παραλαβής του είναι αρκετά μεγάλος και γι'αυτό μερικές φορές απαιτείται μεγάλη διαδικασία σύνδεσης.

1.2.5 Anonymous FTP

Το FTP επέτρεψε σε μια νέα κατηγορία χρηστών να εισέρχονται (login) στον FTP server. Οι χρήστες αυτοί δεν χρειάζεται να έχουν λογαριασμό, αντιθέτως χρησιμοποιούν έναν γενικής χρήσης. Ο λογαριασμός αυτός ονομάζεται "anonymous FTP" και δεν απαιτείται κωδικός πρόσβασης, συνήθως όμως χρησιμοποιείται κατά σύμβαση ο κωδικός "guest" ή η διεύθυνση ηλεκτρονικού ταχυδρομείου (e-mail) του χρήστη.

Η σύνδεση "anonymous" χρησιμοποιείται κυρίως για αρχεία που είναι ανοιχτά στο κοινό, σαν αποθήκη πληροφοριών (όπως λογισμικό, έγγραφα, εικόνες κλπ.) Συνήθως, με αυτό τον τρόπο παρέχεται πρόσβαση σε αρχειοθετημένες mailing lists. Οι χρήστες που εισέρχονται ως "anonymous" πρέπει να έχουν περιορισμένα δικαιώματα πρόσβασης σε

αρχεία του host. Διαφορετικά, εαν μπορούν να διαβάσουν οποιοδήποτε αρχείο ή να δημιουργήσουν νέα, δημιουργούνται προβλήματα ασφαλείας.

1.2.6 Ασφάλεια

Το FTP δεν σχεδιάστηκε με πρόνοια για ασφάλεια, με συνέπεια οι εφαρμογές να είναι ιδιαίτερα ευάλωτες και να εμφανίζονται ποικίλα προβλήματα κατά τη χρήση firewall ή NAT.

1.2.6.1 Προβλήματα NAT

Στην ενεργητική λειτουργία ο FTP server ξεκινά μια σύνδεση δεδομένων συνδεδεμένος στην εξωτερική διεύθυνση IP της πύλης (gateway) NAT. Στην άλλη πλευρά, το μηχάνημα το οποίο είναι υπεύθυνο για τη "μετάφραση" των εσωτερικών διευθύνσεων IP του δικτύου στην εξωτερική, θα πάρει το SYN πακέτο για τη δημιουργία της σύνδεσης. Όμως, στον πίνακα κατάστασης (state table) του NAT, στον οποίο διατηρείται το ιστορικό μεταφράσεων, δεν έχει καταγραφεί κανένα, με αποτέλεσμα το πακέτο να απορρίπτεται (γίνεται drop). Το πακέτο δεν φτάνει ποτέ στον client, δεν σχηματίζεται σύνδεση δεδομένων και η μεταφορά δεδομένων είναι αδύνατη.

Στην παθητική λειτουργία, επειδή η θύρα στην οποία συνδεέται ο server είναι τυχαία, είναι πιθανόν να μην επιτρέπεται σύνδεση προς τον αριθμό της από το λογισμικό - τείχος προστασίας (firewall). Σε αυτή την περίπτωση η σύνδεση δεδομένων δεν θα σχηματιστεί και, επομένως, δεν θα μεταφέρονται δεδομένα.

1.2.6.2 Έλλειψη κρυπτογράφησης

Τα δεδομένα που ανταλλάσσονται μέσω FTP δεν είναι κρυπτογραφημένα, με αποτέλεσμα οι εντολές που αποστέλλονται μέσω της control connection να είναι απλό κείμενο. Για το λόγο αυτό μπορούν εύκολα, με τη χρήση ενός sniffer, να αλιευθούν, να διαβασθούν και να ξανασταλούν ανάλογα με τη βούληση του επιτιθέμενου. Ανάμεσα σε αυτές, η εντολή που χρησιμοποιείται για να γίνει login σε ένα λογαριασμό FTP, με σύνταξη "PASS password", παρέχει στον επιτιθέμενο τον κωδικό του χρήστη. Αν συνδυαστεί με την εντολή "USER", με την οποία αποστέλεται το όνομα του χρήστη, ο επιτιθέμενος μπορεί να χρησιμοποιήσει τα στοιχεία για να εισέλθει στον ξένο λογαριασμό με τα ίδια δικαιώματα.

Επειδή οι περισσότεροι άνθρωποι τείνουν να επαναχρησιμοποιούν κωδικούς, ο επιτιθέμενος έχει αυξήσει τις πιθανότητες του σε μια brute-force attack. Με αυτό τον τρόπο, είναι πιθανό να αποκτήσει έλεγχο του συστήματος του χρήστη μόλις βρεί τη διεύθυνση IP του, ανιχνεύοντας την έναρξη της συνόδου FTP (FTP session).

1.2.6.3 Man-in-the-middle

Με το κύριο File Transfer Protocol, ο server δεν εξασφαλίζει ότι ο client είναι αυτός που λέει, ούτε ο client αντίστοιχα για τον server. Ευκολονόητο, εφόσον το FTP δεν απαιτεί επαλήθευση των hosts και δεν ελέγχει αν τα δεδομένα προέρχονται από αυτούς, ούτε τα

προστατεύει. Για αυτό το λόγο και τα δύο άκρα που ανταλλάσσουν δεδομένα, είναι ανοιχτά σε man-in-the-middle attack από κάποιον επιτιθέμενο που συλλαμβάνει τα πακέτα του κάθε host, και στέλνει ψευδείς απαντήσεις. (Πηγή: http://el.wikipedia.org/wiki/File_Transfer_Protocol)

Κεφάλαιο 2

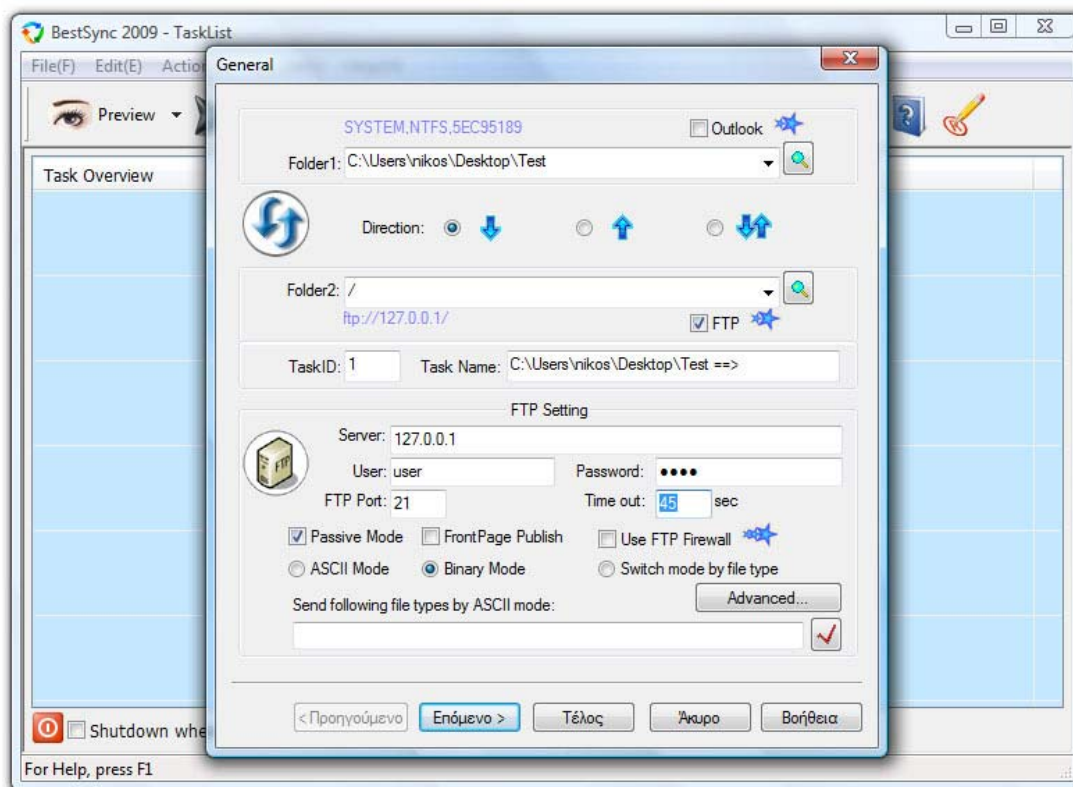
2.1 Γενικά

Στο κεφάλαιο αυτό θα γίνει η ανάλυση του τρόπου λειτουργίας μερικών γνωστών εφαρμογών συγχρονισμού ενός φακέλου του τοπικού υπολογιστή με ένα φάκελο ενός απομακρυσμένου διακομιστή FTP. Μετά το πέρας της ανάλυσης θα γίνει σύγκριση των εφαρμογών αυτών με την εφαρμογή της παρούσας πτυχιακής.

2.2 BestSynch

Το BestSynch είναι ένα πολυπρόγραμμα το οποίο επιτρέπει το συγχρονισμό μεταξύ δυο φακέλων του τοπικού υπολογιστή, ενός φακέλου του τοπικού υπολογιστή με ένα φάκελο του δικτύου ή ενός FTP Server ή ακόμα ενός αφαιρούμενου δίσκου. Στην παρούσα εργασία θα αναλύσουμε μόνο την περίπτωση συγχρονισμού ενός φακέλου του τοπικού υπολογιστή με ένα φάκελο ενός FTP Server, η οποία μας ενδιαφέρει.

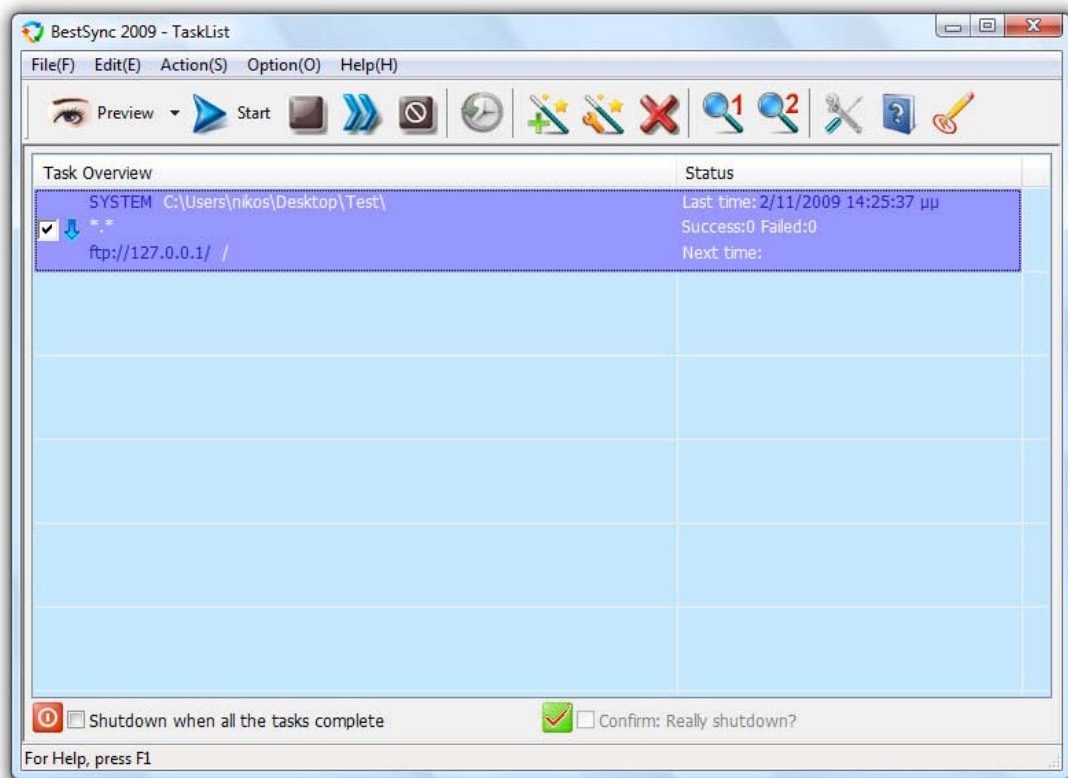
Το πρόγραμμα λειτουργεί με τη δημιουργία μιας λίστας προγραμματισμένων εργασιών συγχρονισμού (Tasks). Οι προσθήκη μιας προγραμματισμένης εργασίας γίνεται μέσω ενός οδηγού προσθήκης εργασιών (Add Task Wizard).



Εικόνα 2.1 Ο οδηγός προσθήκης εργασίας του BestSynch

Ο οδηγός προσθήκης εργασίας αποτελείται από τα πεδία στα οποία συμπληρώνει ο χρήστης τα στοιχεία της σύνδεσης με τον FTP Server όπως τη διεύθυνση του FTP Server (*Server*), το όνομα του χρήστη (*User*), τον κωδικό (*Password*), τη θύρα με την οποία θα συνδεθεί (*FTP Port*) καθώς επίσης και το μέγιστο χρόνο που θα επιτρέπεται σε μια σύνδεση να είναι ανοιχτή (*Timeout*). Επίσης δίνεται στον χρήστη η δυνατότητα της επιλογής του φακέλου του τοπικού υπολογιστή που θα συγχρονιστεί με τον φάκελο του FTP Server, συμπληρώνοντας το πεδίο επιλογής φακέλου (*Folder 1*), καθώς και με ποίο φάκελο του FTP Server θα συγχρονιστεί, συμπληρώνοντας το πεδίο επιλογής φακέλου (*Folder 2*). Η επιλογή των φακέλων μπορεί να γίνει είτε συμπληρώνοντας το μονοπάτι του κάθε φακέλου στο αντίστοιχο πεδίο είτε πατώντας τα πλήκτρα δίπλα από τα πεδία και επιλέγοντας τον επιθυμητό φάκελο μέσω του Browser. Μια ακόμα από τις επιλογές που διαθέτει ο χρήστης, μέσω του πεδίου επιλογής κατεύθυνσης συγχρονισμού (*Direction*), είναι αν ο συγχρονισμός θα είναι μονόδρομος, δηλαδή οι αλλαγές μόνο του επιλεγμένου φακέλου να γίνονται στον άλλο, ή αμφίδρομος, δηλαδή οι αλλαγές που γίνονται σε οποιοδήποτε από τους δυο φακέλους να γίνονται και στον άλλο.

Ο χρήστης μπορεί να επιλέξει τη θέση της εργασίας στη λίστα προγραμματισμένων εργασιών, αλλάζοντας την τιμή στο πεδίο *Task ID*, καθώς και το όνομα αυτής, τοποθετώντας το επιθυμητό όνομα στο πεδίο *Task Name*.



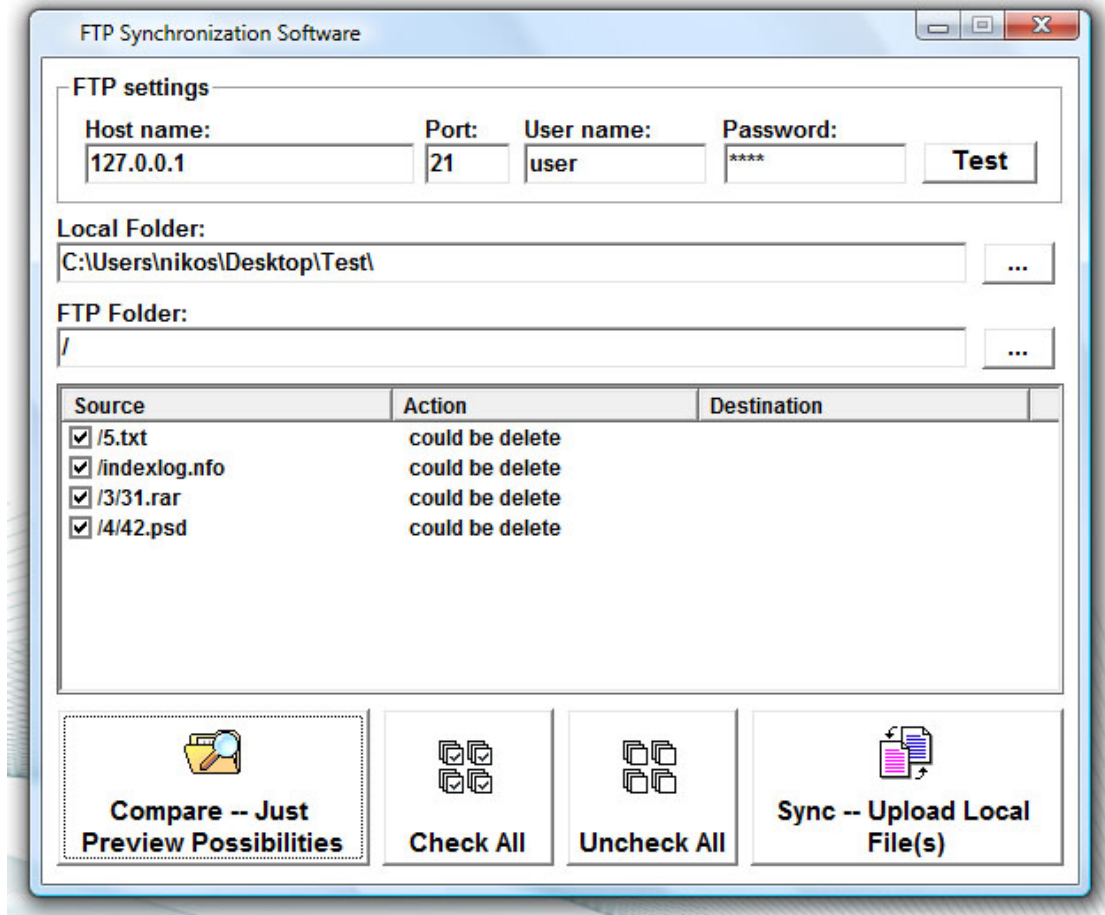
Εικόνα 2.2 Η λίστα προγραμματισμένων εργασιών του BestSynch

Ο χρήστης στην αρχική σελίδα του προγράμματος να επιλέξει την εκτέλεση της επιλεγμένης εργασίας, χρησιμοποιώντας την επιλογή *Start Task*, ή όλων των τσεκαρισμένων εργασιών της λίστας, χρησιμοποιώντας την επιλογή *Start All Checked Tasks*. Η διακοπή μιας εργασίας συγχρονισμού που εκτελείται γίνεται με την επιλογή *Stop Task* ή με την επιλογή *Stop All Tasks*, με την οποία διακόπτονται όλες οι εργασίες που εκτελούνται. Επίσης δίνεται η δυνατότητα στον χρήστη να δει μια λίστα προεπισκόπησης όλων των αλλαγών που πρόκειται να γίνουν σε μία εργασία, χρησιμοποιώντας την επιλογή *Preview*.

Ο χρήστης μπορεί να προσθέσει διάφορες εργασίες στη λίστα, χρησιμοποιώντας την επιλογή *Add Task* (η οποία ανοίγει τον οδηγό προσθήκης εργασιών), να τροποποιήσει μια υπάρχουσα εργασία, χρησιμοποιώντας την επιλογή *Modify Task*, ή να διαγράψει μια εργασία από τη λίστα χρησιμοποιώντας την επιλογή *Delete Task*. Επίσης μπορεί να περιηγηθεί στους φακέλους μιας επιλεγμένης εργασίας χρησιμοποιώντας τις επιλογές *Explore Folder 1* και *Explore Folder 2* αντίστοιχα.

2.3 FTP Synchronization Software

Το FTP Synchronization Software είναι ένα πρόγραμμα συγχρονισμού ενός φακέλου του τοπικού υπολογιστή με ένα φάκελο ενός FTP Server.



Εικόνα 2.3 Το πρόγραμμα FTP Synchronization Software

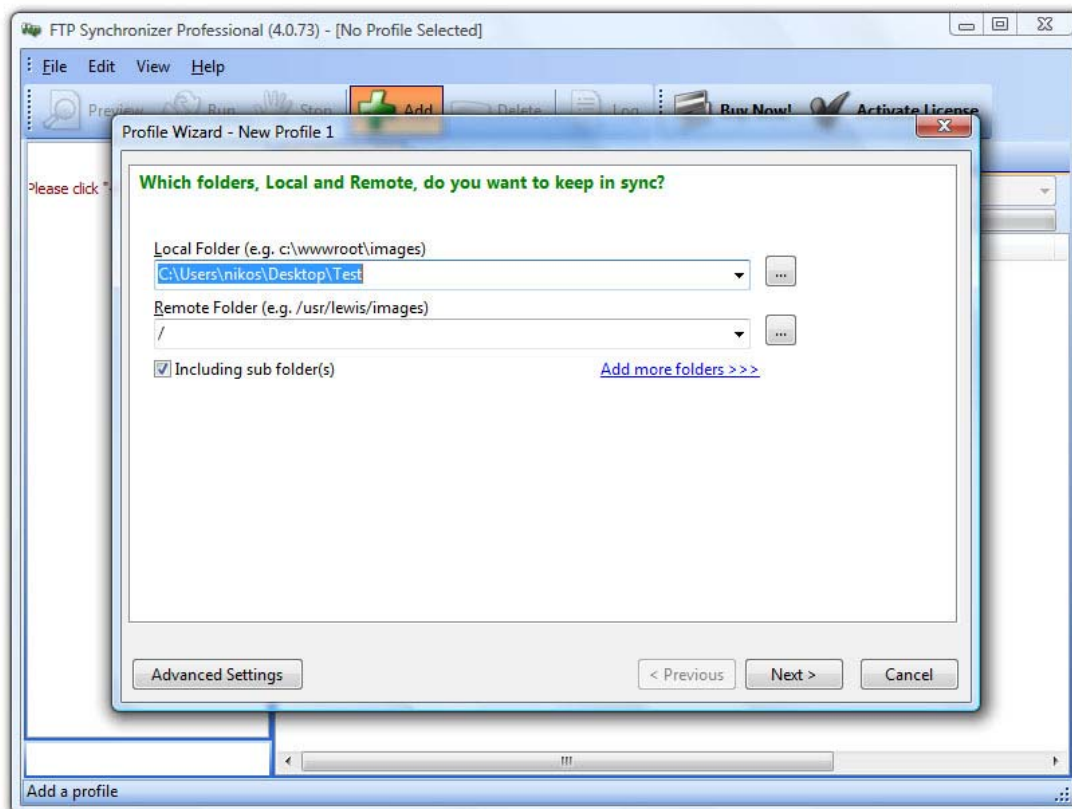
Το πρόγραμμα αποτελείται από τα πεδία στα οποία συμπληρώνει ο χρήστης τα στοιχεία της σύνδεσης με τον FTP Server όπως τη διεύθυνση του FTP Server (*Host name*), τη θύρα με την οποία θα συνδεθεί (*Port*), το όνομα του χρήστη (*Username*) και τον κωδικό (*Password*). Επίσης δίνεται στον χρήστη η δυνατότητα της επιλογής του φακέλου του τοπικού υπολογιστή που θα συγχρονιστεί με τον φάκελο του FTP Server, συμπληρώνοντας το πεδίο επιλογής φακέλου (*Local folder*), καθώς και με ποιο φάκελο του FTP Server θα συγχρονιστεί, συμπληρώνοντας το πεδίο επιλογής φακέλου (*FTP folder*). Η επιλογή των φακέλων μπορεί να γίνει είτε συμπληρώνοντας το μονοπάτι του κάθε φακέλου στο αντίστοιχο πεδίο είτε πατώντας τα πλήκτρα δίπλα από τα πεδία και επιλέγοντας τον επιθυμητό φάκελο μέσω του Browser.

Με την ορθή συμπλήρωση των προαναφερθέντων πεδίων ο χρήστης, πατώντας το πλήκτρο *Compare*, μπορεί να δει μία λίστα με όλα τα διαθέσιμα προς συγχρονισμό αρχεία και να επιλέξει τα αρχεία που επιθυμεί για συγχρονισμό. Επίσης του δίνεται η δυνατότητα της αυτόματης επιλογής όλων των αρχείων της λίστας, πατώντας το πλήκτρο *Check All*, ή πατώντας το πλήκτρο *Uncheck All* να μην επιλέξει κανένα. Τέλος, πατώντας το πλήκτρο *Sync* γίνεται ο συγχρονισμός των δυο φακέλων με τα αρχεία της λίστας που επιλέχθηκαν.

2.4 FTP Synchronizer Professional

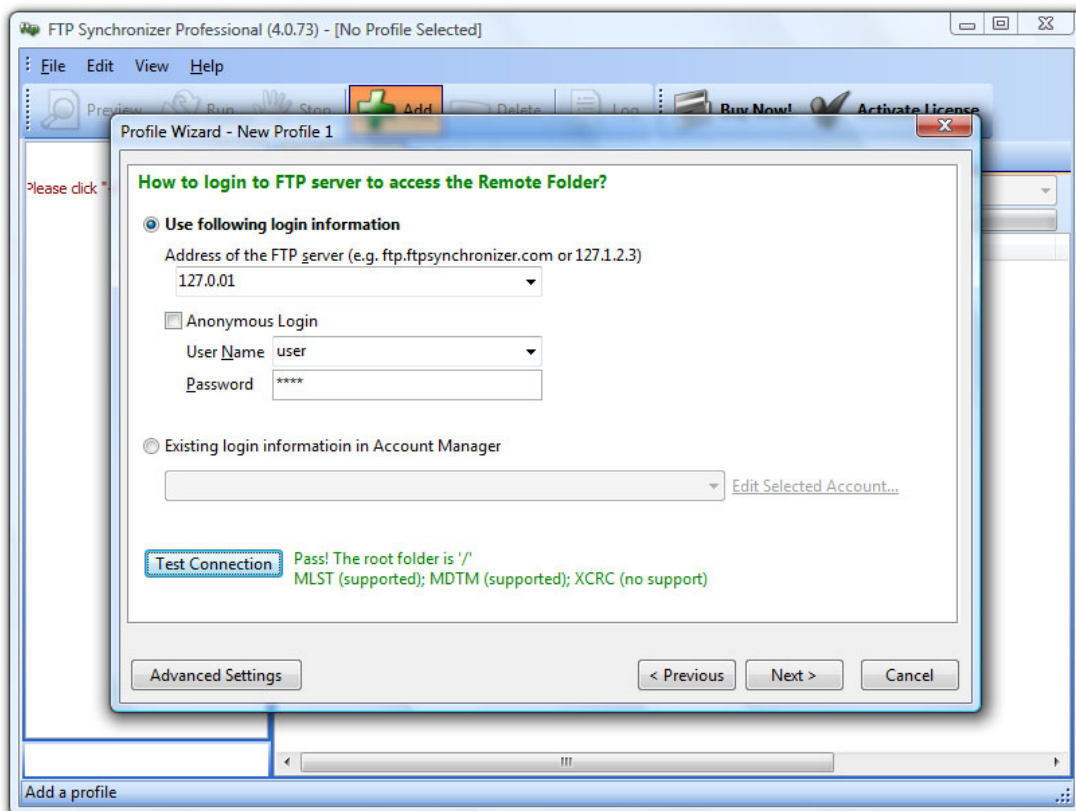
Το FTP Synchronizer Professional είναι ένα πρόγραμμα συγχρονισμού ενός φακέλου του τοπικού υπολογιστή με ένα φάκελο ενός FTP Server.

Το πρόγραμμα λειτουργεί με τη δημιουργία μιας λίστας με προφίλ χρηστών. Κάθε προφίλ της λίστας αντιπροσωπεύει ένα διαφορετικό ζευγάρι φακέλων συγχρονισμού (φάκελος τοπικού υπολογιστή – φάκελος FTP Server). Η προσθήκη των διάφορων προφίλ στη λίστα γίνεται με τη χρήση ενός οδηγού προσθήκης προφίλ (Add Profile Wizard) χρησιμοποιώντας την επιλογή *Add* του προγράμματος.



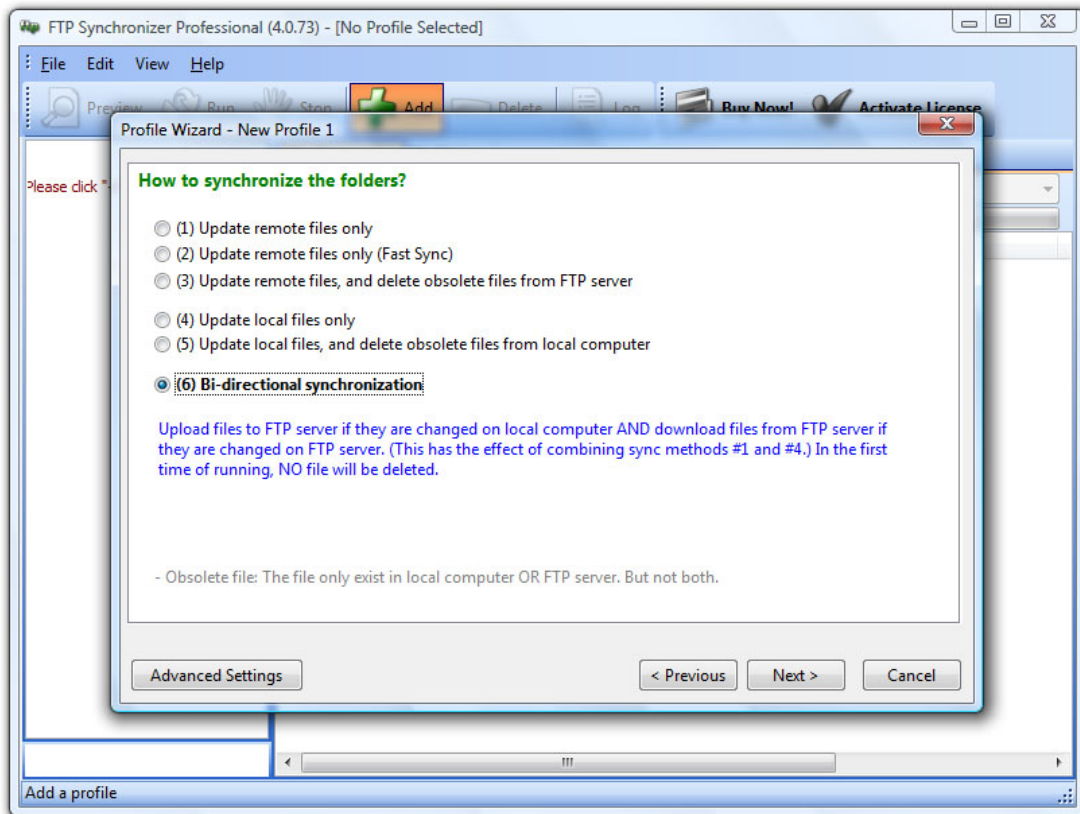
Εικόνα 2.4 Η σελίδα επιλογής φακέλων του οδηγού προσθήκης προφίλ

Ο οδηγός προσθήκης προφίλ αποτελείται από πέντε μέρη. Στο πρώτο μέρος δίνεται στον χρήστη η δυνατότητα της επιλογής του φακέλου του τοπικού υπολογιστή που θα συγχρονιστεί με τον φάκελο του FTP Server, συμπληρώνοντας το πεδίο επιλογής φακέλου (*Local Folder*), καθώς και με ποίο φάκελο του FTP Server θα συγχρονιστεί, συμπληρώνοντας το πεδίο επιλογής φακέλου (*Remote Folder*). Η επιλογή των φακέλων μπορεί να γίνει είτε συμπληρώνοντας το μονοπάτι του κάθε φακέλου στο αντίστοιχο πεδίο είτε πατώντας τα πλήκτρα δίπλα από τα πεδία και επιλέγοντας τον επιθυμητό φάκελο μέσω του Browser.



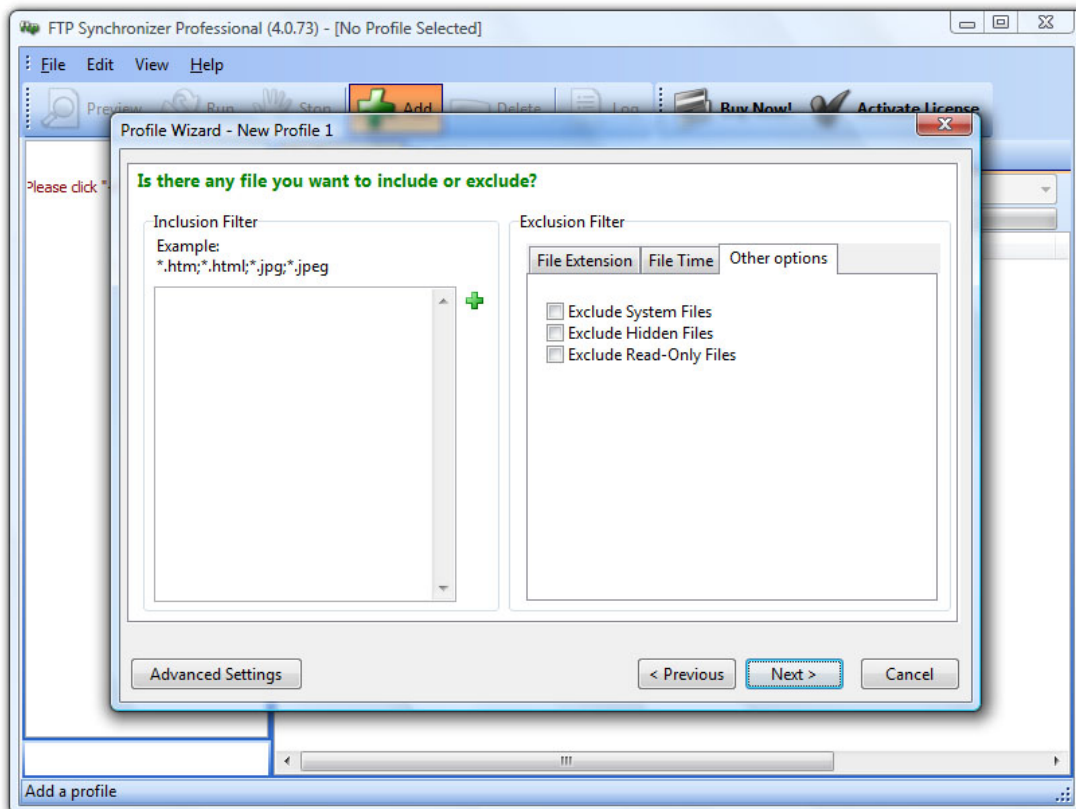
Εικόνα 2.5 Η σελίδα επιλογής FTP Server του οδηγού προσθήκης προφίλ

Το δεύτερο μέρος του οδηγού αποτελείται από τα πεδία στα οποία συμπληρώνει ο χρήστης τα στοιχεία της σύνδεσης με τον FTP Server όπως τη διεύθυνση του FTP Server (*Address of the FTP server*), το όνομα του χρήστη (*User Name*) και τον κωδικό (*Password*). Ο χρήστης μπορεί να ελέγξει την ορθότητα των στοιχείων, αλλά και την ύπαρξη της σύνδεσης γενικά, δημιουργώντας μια σύνδεση με τον FTP Server, πατώντας το κουμπί *Test Connection*.



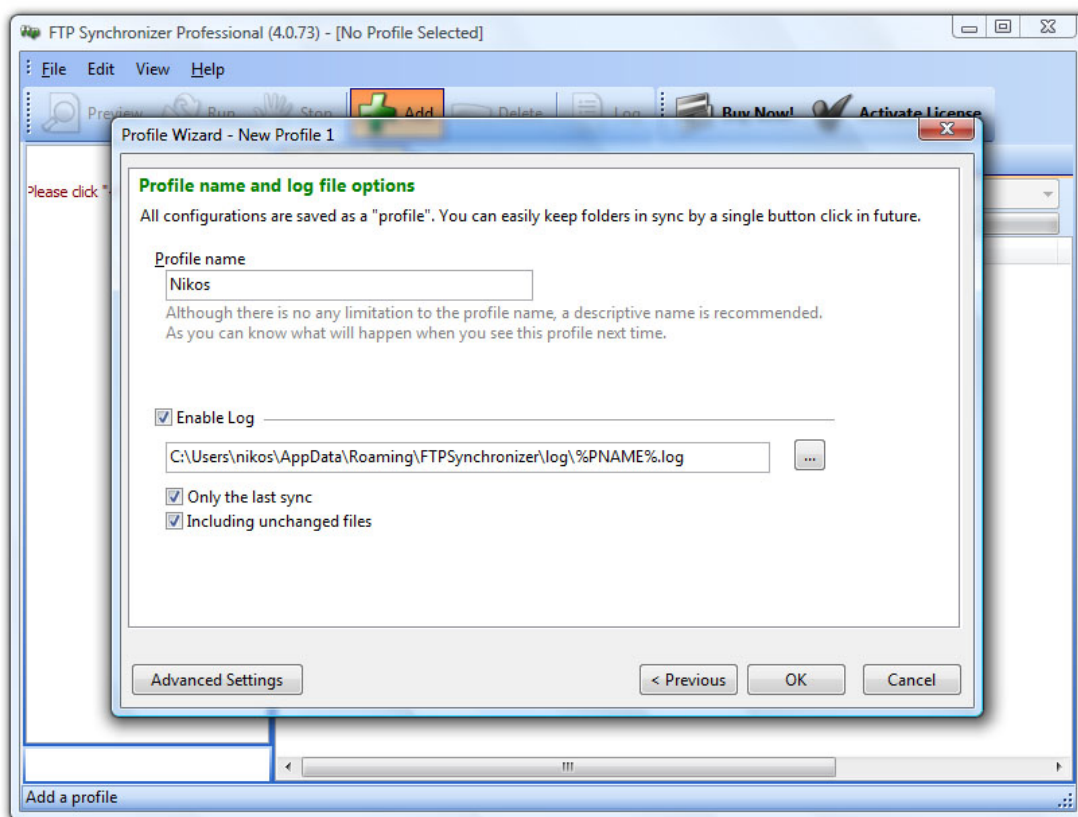
Εικόνα 2.6 Η σελίδα επιλογής μεθόδων συγχρονισμού του οδηγού προσθήκης προφίλ

Στο τρίτο μέρος του οδηγού ο χρήστης επιλέγει αν ο συγχρονισμός μεταξύ των δυο φακέλων θα είναι μονόδρομος, δηλαδή οι αλλαγές μόνο του επιλεγμένου φακέλου να γίνονται στον άλλο, ή αμφίδρομος, δηλαδή οι αλλαγές που γίνονται σε οποιοδήποτε από τους δυο φακέλους να γίνονται και στον άλλο.



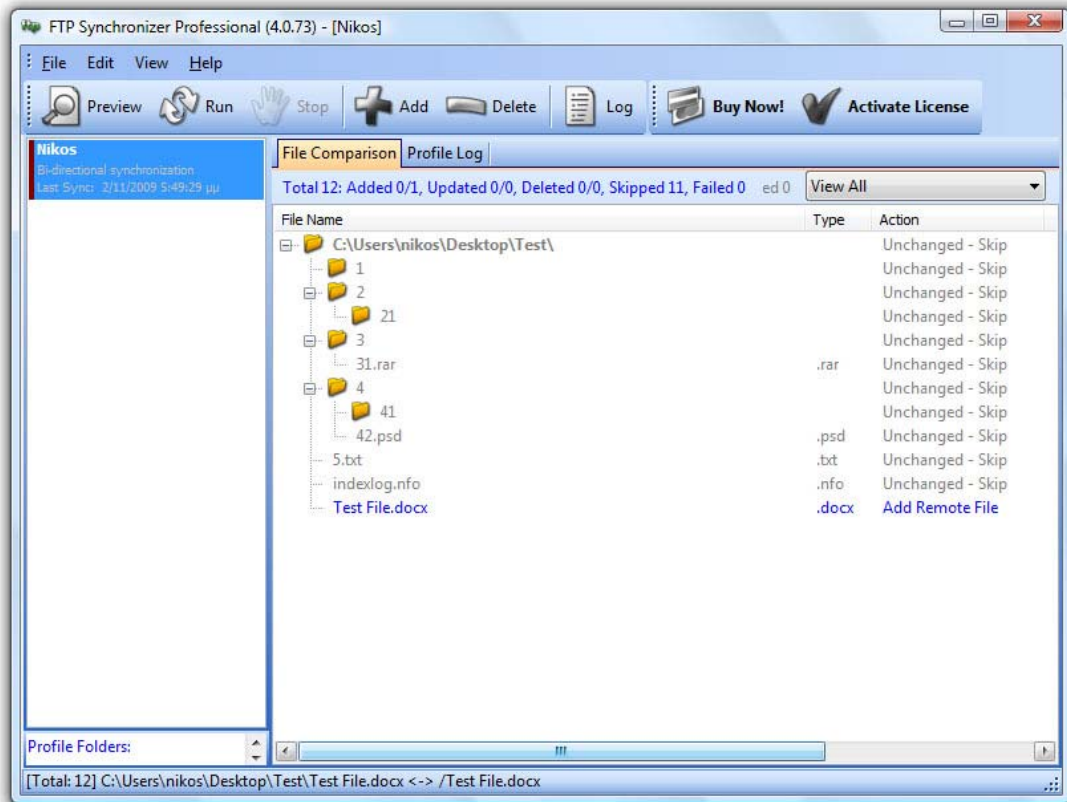
Εικόνα 2.7 Η σελίδα προσθήκης εξαιρέσεων του οδηγού προσθήκης προφίλ

Στο τέταρτο μέρος του οδηγού ο χρήστης επιλέγει κάποια φίλτρα προσθήκης συγκεκριμένων τύπων αρχείων (*Inclusion Filter*) ή απόρριψης κάποιων αρχείων (*Exclusion Filter*) βάσει του τύπου του αρχείου ή του ονόματος του (*File Extensions*), της παλαιότητας του (*File Time*) ή κάποιων ιδιοτήτων του (*Other Options*).



Εικόνα 2.8 Η σελίδα επιλογής ονόματος του οδηγού προσθήκης προφίλ

Στο πέμπτο, και τελευταίο, μέρος του οδηγού ο χρήστης επιλέγει το όνομα του προφίλ, συμπληρώνοντας το στον πεδίο *Profile name*, και αν επιθυμεί τη δημιουργία ενός αρχείου καταγραφής ενεργειών του προγράμματος καθώς και το φάκελο στον οποίο επιθυμεί να αποθηκεύεται το αρχείο, συμπληρώνοντας το αντίστοιχο πεδίο του *Enable Log*.



Εικόνα 2.9 Το FTP Synchronizer Professional

Μετά την επιτυχημένη δημιουργία του προφίλ του χρήστη το πρόγραμμα εμφανίζει σε μια λίστα όλα τα αρχεία του φακέλου του τοπικού υπολογιστή. Ο χρήστης, χρησιμοποιώντας την επιλογή *Preview*, δίνει εντολή στο πρόγραμμα να ελέγξει τους δύο προς συγχρονισμό φακέλους και να ανανεώνει τη λίστα αρχείων δείχνοντας τι ενέργεια θα πραγματοποιηθεί για κάθε αρχείο ξεχωριστά (κατέβασμα αρχείου στον τοπικό φάκελο, αποστολή αρχείου στον FTP φάκελο, διαγραφή αρχείου από τον FTP φάκελο κ.ο.κ.). Έπειτα, χρησιμοποιώντας την επιλογή *Run*, ο χρήστης δίνει εντολή στο πρόγραμμα να πραγματοποιήσει αυτές τις αλλαγές και να συγχρονίσει τους δύο φακέλους.

2.5 Συγκριση της πτυχιακής

Ο κοινός στόχος όλων των προαναφερθέντων προγραμμάτων και της παρούσας πτυχιακής είναι ένας: ο συγχρονισμός ενός φακέλου του τοπικού υπολογιστή με το φάκελο που βρίσκεται στον απομακρυσμένο διακομιστή FTP. Με αυτόν τον τρόπο καταφέρνουμε να έχουμε ανα πάσα στιγμή διαθέσιμα τα αρχεία που έχουμε επιλέξει σε οποιοδήποτε υπολογιστή κι αν βρισκόμαστε.

Εκτός όμως από τις ομοιότητες υπάρχουν και διαφορές. Μια βασική διαφορά είναι ότι όλα τα προαναφερθέντα προγράμματα για να λειτουργήσουν απαιτούν την εγκατάστασή τους (install) στον τοπικό υπολογιστή. Η απαίτηση αυτή τα καθιστά ακατάλληλα για χρήση

σε υπολογιστές που ο χρήστης έχει περιορισμένα δικαιώματα, όπως για παράδειγμα οι υπολογιστές των Internet Cafeé. Στην εφαρμογή μας το μόνο που απαιτείται είναι ο χρήστης να έχει το δικαίωμα να τρέχει εκτελέσιμα αρχεία.

Η αναγκαστική εκτέλεση κάποιων ενεργειών για τη σωστή λειτουργία του προγράμματος, όπως για παράδειγμα η δημιουργία προφίλ μέσω ενός οδηγού δημιουργίας προφίλ του προγράμματος FTP Synchronizer Professional, μπορεί να κουράσει ορισμένους χρήστες οι οποίοι θα επιθυμούν την ολοκλήρωση του συγχρονισμού με όσο το δυνατόν λιγότερα βήματα. Στην εφαρμογή μας το μόνο που απαιτείται από τον χρήστη είναι η συμπλήρωση των απαραίτητων στοιχείων για τη σύνδεση με τον απομακρυσμένο διακομιστή FTP (διεύθυνση FTP Server, όνομα χρήστη και κωδικός πρόσβασης).

Επίσης η επιλογή των προς συγχρονισμό αρχείων μέσα από μια λίστα μπορεί να αποθαρρύνει μερικούς, όχι και τόσο εξοικωμένους, χρήστες από τη χρησιμοποίηση του προγράμματος. Στην εφαρμογή μας ο χρήστης διαχειρίζεται τα περιεχόμενα του φακέλου που δημιουργεί η εφαρμογή όπως θα διαχειριζόταν ένα οποιοδήποτε αρχείο ή φάκελο, τοποθετώντας τα αρχεία που επιθυμεί μέσα στο φάκελο και διαγράφοντας αυτά που δεν επιθυμεί.

Ένα από τα βασικά ελλείμματα της εφαρμογής μας είναι ότι η διαχείριση του φακέλου που βρίσκεται στον απομακρυσμένο διακομιστή FTP πρέπει να γίνεται αποκλειστικά και μόνο από την εφαρμογή. Οποιαδήποτε αλλαγή γίνει στο φάκελο του απομακρυσμένου διακομιστή FTP από οποιοδήποτε άλλο πρόγραμμα, η εφαρμογή μας δεν θα την αναγνωρίσει με αποτέλεσμα να μη μπορούμε να κάνουμε σωστή διαχείριση των αρχείων.

Κεφάλαιο 3

3.1.Γενικά

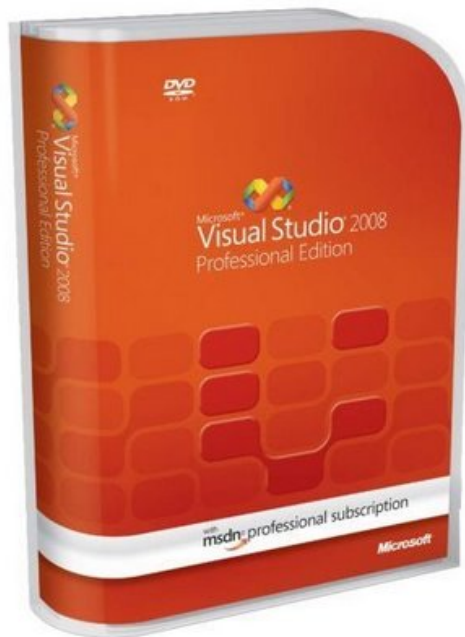
Όπως προαναφέρθηκε για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το πρόγραμμα Microsoft Visual Studio 2008 με γλώσσα προγραμματισμού την Visual Basic. Στο κεφάλαιο αυτό θα αναλύσουμε την αρχιτεκτονική και τον τρόπο λειτουργίας του Microsoft Visual Studio, της γλώσσας προγραμματισμού Microsoft Visual Basic και το .NET Framework της Microsoft.

3.2 Microsoft Visual Studio

Το Microsoft Visual Studio αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) από τη Microsoft. Μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών κονσόλας, γραφικών εφαρμογών, ιστοσελίδων, εφαρμογών Web και υπηρεσιών web για όλες τις πλατφόρμες που υποστηρίζονται από τα Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework και Microsoft Silverlight.

Το Microsoft Visual Studio περιλαμβάνει έναν συντάκτη κώδικα που υποστηρίζει IntelliSense καθώς επίσης και κώδικα. Ο ενσωματωμένος διορθωτής (debugger) λειτουργεί και σαν διορθωτής πηγής (source-level debugger) αλλά και σαν διορθωτής μηχανής (machine-level debugger). Άλλα ενσωματωμένα εργαλεία περιλαμβάνουν έναν σχεδιαστή φορμών για την σχεδίαση των γραφικών εφαρμογών (Graphical User Interface - GUI), το σχεδιαστή Ιστού, το σχεδιαστή κλάσεων, και το σχεδιαστή σχημάτων βάσεων δεδομένων.

Το Microsoft Visual Studio υποστηρίζει τις γλώσσες, με τη έννοια των γλωσσικών υπηρεσιών, οι οποίες επιτρέπουν σε οποιαδήποτε γλώσσα προγραμματισμού για να υποστηριχθούν (σε διαφορετικό βαθμό) από το συντάκτη και το διορθωτή κώδικα, υπό τον όρο ότι μια συγκεκριμένη γλωσσικά υπηρεσία ήταν. Στις ενσωματωμένες γλώσσες περιλαμβάνονται οι C/C++ (μέσω Visual C++), VB.NET (μέσω Visual Basic .NET), και C# (μέσω Visual C#). Η υποστήριξη για άλλες γλώσσες όπως F#, το M, Python, και την Ruby μεταξύ των άλλων έχει παρασχεθεί μέσω των γλωσσικών υπηρεσιών που πρόκειται να εγκατασταθούν χωριστά. Υποστηρίζει επίσης XML/XSLT, HTML/XHTML, JavaScript και CSS.



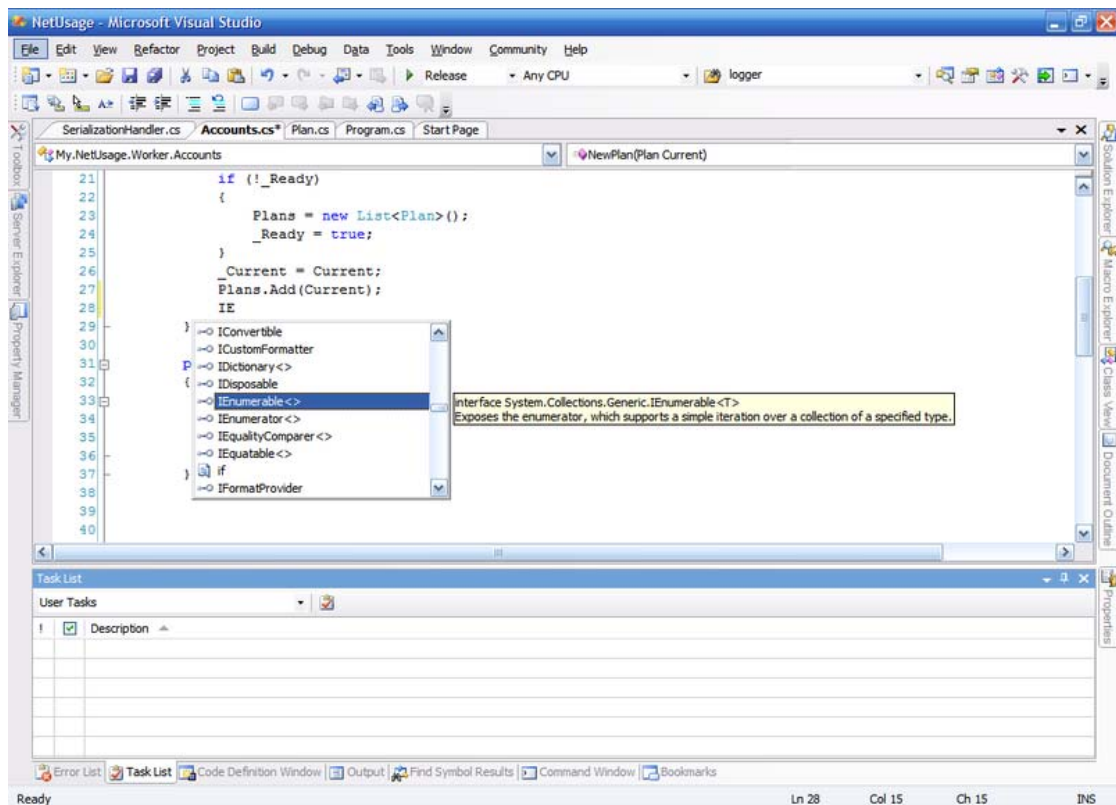
Εικόνα 3.1 Το Microsoft Visual Studio

3.2.1 Χαρακτηριστικά Γνωρίσματα

3.2.1.1 Συντάκτης Κώδικα

Το Microsoft Visual Studio, όπως οποιοδήποτε άλλο IDE, περιλαμβάνει έναν συντάκτη κώδικα που υποστηρίζει τον τονισμό σύνταξης και την ολοκλήρωση κώδικα, χρησιμοποιώντας την τεχνολογία IntelliSense, όχι μόνο για τις μεταβλητές, τις λειτουργίες και τις μεθόδους αλλά και τα γλωσσικά κατασκευάσματα όπως τους βρόχους και τις ερωτήσεις. Η λειτουργία IntelliSense υποστηρίζεται για τις συμπεριλαμβανόμενες γλώσσες, καθώς επίσης και για XML, Cascading Style Sheets (CSS) και JavaScript κατά την ανάπτυξη των ιστοχώρων (web sites) και των εφαρμογών web (web applications). Οι προτάσεις για την αυτόματη συμπλήρωση εμφανίζονται πάνω σε ένα παράθυρο καταλόγων, πάνω από το συντάκτη κώδικα, το οποίο γίνεται προσωρινά ημιδιάφανο για να διευκολύνει τον χρήστη να βλέπει τον κώδικα.

Ένα από τα χαρακτηριστικά του Microsoft Visual Studio είναι η σύνταξη υποβάθρου (background compilation). Καθώς γράφεται ο κώδικας, το Microsoft Visual Studio ελέγχει για την ορθή σύνταξη στο υπόβαθρο. Τα λάθη σύνταξης υπογραμμίζονται με μια κόκκινη κυματιστή γραμμή, ενώ οι προειδοποιήσεις με μια πράσινη κυματιστή γραμμή. Η σύνταξη υποβάθρου δεν παράγει τον εκτελέσιμο κώδικα, δεδομένου ότι χρησιμοποιεί έναν διαφορετικό μεταγλωττιστή από αυτόν που χρησιμοποιείται για την παραγωγή του εκτελέσιμου κώδικα. Η σύνταξη υποβάθρου εισήχθη αρχικά στη Microsoft Visual Basic αλλά έχει πλέον επεκταθεί σε όλες τις συμπεριλαμβανόμενες γλώσσες.



Εικόνα 3.2 Η Τεχνολογία IntelliSense του Microsoft Visual Studio

3.2.1.2 Διορθωτής

Το Microsoft Visual Studio περιλαμβάνει έναν διορθωτή (debugger) που χρησιμοποιείται και ως διορθωτής πηγής (source-level debugger) και ως διορθωτής μηχανής (machine-level debugger). Ο διορθωτής χρησιμοποιείται για την διόρθωση των εφαρμογών που γράφονται σε οποιαδήποτε γλώσσα που υποστηρίζεται από το Microsoft Visual Studio. Επιπλέον, μπορεί επίσης να συνδεθεί με τις τρέχουσες διεργασίες και να επιτηρεί και να επιδιορθώνει τις διεργασίες αυτές.

Ο διορθωτής επιτρέπει τα σημεία παύσης (breakpoints), που επιτρέπουν στο πρόγραμμα να σταματήσει προσωρινά σε μια ορισμένη θέση κατά την εκτέλεση, και τα ρολόγια, τα οποία ελέγχουν τις τιμές των μεταβλητών καθώς η εκτέλεση προχωρεί. Τα σημεία παύσης μπορούν να είναι υπό όρους, που σημαίνει ότι προκαλούνται μόνο όταν ο όρος ικανοποιείται. Ο διορθωτής υποστηρίζει την λειτουργία Τροποποίηση και Συνέχεια (Edit And Continue), το οποίο επιτρέπει στον χρήστη να τροποποιεί τον κώδικα κατά τη διάρκεια της διόρθωσης.

3.2.1.3 Σχεδιαστής

Το Microsoft Visual Studio περιλαμβάνει έναν πλήθος από οπτικούς σχεδιαστές (Visual Designers) οι οποίοι βοηθούν στην ανάπτυξη των εφαρμογών. Αυτά τα εργαλεία περιλαμβάνουν:

- Microsoft Visual Studio 2005 στην Designer View
- Ο σχεδιαστής WPF στο Microsoft Visual Studio 2008
- Visual Studio σχεδιαστής Ιστού (Web Designer) Code Editor View
- Microsoft Visual Studio 2005 Class Designer View

Σχεδιαστής μορφών παραθύρων

Ο σχεδιαστής μορφών παραθύρων (Windows Form Designer – WFD) χρησιμοποιείται για τη δημιουργία γραφικών εφαρμογών. Περιλαμβάνει μια παλέτα που αποτελείται από αντικείμενα του User Interface (UI) και διάφορους ελεγκτές (συμπεριλαμβανομένων των κουμπιών, των κουτιών επιλογής, των ετικετών και άλλων ελεγκτών) που μπορεί να τοποθετηθούν σε ένα παράθυρο με την μέθοδο Drag And Drop. Οι έλεγχοι που επιδεικνύουν στοιχεία, όπως το Textbox ή το Listbox, μπορούν να παίρνουν αυτά τα στοιχεία από διάφορες πηγές όπως, για παράδειγμα, τις βάσεις δεδομένων.

Σχεδιαστής WPF

Ο WPF σχεδιαστής, με κωδικό Cinder (Μηλίτης), εισήχθη με την έκδοση Microsoft Visual Studio 2008. Όπως στο σχεδιαστή μορφών παραθύρων (WFD) υποστηρίζει τη μεταφορά των αντικειμένων του User Interface (UI) και των διαφόρων ελεγκτών από την παλέτα σε ένα σχεδιαστικό παράθυρο με την μέθοδο Drag And Drop. Χρησιμοποιείται για την δημιουργία γραφικών εφαρμογών απευθυνόμενος στο ίδρυμα παρουσίασης παραθύρων (Windows Presentation Foundation – WPF). Υποστηρίζει όλη τη λειτουργία WPF συμπεριλαμβανομένης και της αυτόματης διαχείρισης σχεδιαγράμματος. Παράγει τον κώδικα XAML για το UI. Το παραγμένο αρχείο XAML είναι συμβατό με το σχέδιο έκφρασης της Microsoft (Microsoft Expression Design). Ο κώδικας XAML συνδέεται με τον κώδικα που χρησιμοποιώντας ένα Code - Behind πρότυπο.

Σχεδιαστής Ιστού

Το Microsoft Visual Studio περιλαμβάνει επίσης έναν συντάκτη και τον αντίστοιχο σχεδιαστή ιστοσελίδας υποστηρίζει τη μεταφορά των αντικειμένων του User Interface (UI) και των διαφόρων ελεγκτών από την παλέτα σε ένα σχεδιαστικό παράθυρο με την μέθοδο Drag And Drop. Χρησιμοποιείται για την ανάπτυξη ASP.NET εφαρμογών και υποστηρίζει το HTML, CSS και JavaScript. Χρησιμοποιεί ένα code behind πρότυπο για να συνδέσει με τον κώδικα ASP.NET. Από το Microsoft Visual Studio το 2008 και μετά, η μηχανή σχεδιασμού που χρησιμοποιείται από το σχεδιαστή Ιστού μοιράζεται με το Microsoft Web Expression.

Σχεδιαστής κλάσεων

Ο σχεδιαστής κλάσεων χρησιμοποιείται στο συντάκτη και τροποποιεί τις κλάσεις (συμπεριλαμβανομένων των μελών τους και της πρόσβασής τους) χρησιμοποιώντας τη διαμόρφωση UML. Ο σχεδιαστής κατηγορίας μπορεί να παραγάγει τις περιλήψεις κώδικα C# και VB.NET για τις κατηγορίες και τις μεθόδους. Μπορεί επίσης να παραγάγει τα διαγράμματα κλάσεων από τις κλάσεις του χρήστη.

Σχεδιαστής στοιχείων

Ο σχεδιαστής στοιχείων μπορεί να χρησιμοποιηθεί για να τροποποιηθούν γραφικά τα διαγράμματα των βάσεων δεδομένων (Data Base Schemas), συμπεριλαμβανομένων των πινάκων της βάσης, τα πρωτεύοντα και δευτερεύοντα κλειδιά καθώς και τους όποιους περιορισμούς.

3.2.1.4 Άλλα Εργαλεία

Ανοικτή μηχανή αναζήτησης ετικετών

Η ανοικτή μηχανή αναζήτησης ετικετών δημιουργεί μια λίστα με όλες τις ανοιχτές ετικέτες του προγράμματος και επιτρέπει την εναλλαγή μεταξύ αυτών, χρησιμοποιώντας τον συνδυασμό των πλήκτρων CTRL+TAB.

Συντάκτης ιδιοτήτων

Το εργαλείο *συντακτών ιδιοτήτων* δημιουργεί μια λίστα με όλες τις διαθέσιμες ιδιότητες (όχι μόνο αυτές που μπορούν να τροποποιηθούν) για όλα τα αντικείμενα, συμπεριλαμβανομένων των κλάσεων, των αντικειμένων μιας φόρμας, των ιστοσελίδων και άλλων στοιχείων, της εκάστοτε εφαρμογής.

Εξερευνητής λύσης

Στο οπτικό ιδίωμα στούντιο, μια λύση είναι ένα σύνολο αρχείων κώδικα και άλλων πόρων που χρησιμοποιούνται για να χτίσουν μια εφαρμογή. Τα αρχεία σε μια λύση τακτοποιούνται ιεραρχικά, η οποία μπορεί (ή όχι) να απεικονίζει την οργάνωση στο σύστημα αρχείων. Ο *εξερευνητής λύσης* χρησιμοποιείται για να διαχειριστεί και να ταξινομήσει τα αρχεία σε μια λύση.

Εξερευνητής κεντρικών υπολογιστών

Το εργαλείο *εξερευνητών κεντρικών υπολογιστών* χρησιμοποιείται για να διαχειριστεί τις συνδέσεις βάσεων δεδομένων σε έναν προσιτό υπολογιστή. Χρησιμοποιείται επίσης για να ταξινομήσει τις τρέχουσες υπηρεσίες των Windows, τους μετρητές απόδοσης, τα αρχεία καταγραφής γεγονότων των Windows καθώς και τα διάφορα μηνύματα της εφαρμογής και να τις χρησιμοποιήσει ως πηγή δεδομένων.

3.2.2 Συμπεριλαμβανόμενα Προϊόντα

3.2.2.1 *Microsoft Visual C++*

Η Microsoft Visual C++ είναι η προσθήκη του μεταγλωττιστή της Microsoft C και C++ και συγκεκριμένων εργαλείων και γλωσσικών υπηρεσιών για την ενσωμάτωση με το Visual Studio IDE. Μπορεί να χρησιμοποιηθεί είτε σαν C είτε σαν C++ μεταγλωττιστής. Σαν C, ακολουθεί τα πρότυπα του ISO C με τα μέρη C99 της προδιαγραφής και με συγκεκριμένες προσθήκες υπό μορφή βιβλιοθηκών. Σαν C++, ακολουθεί την προδιαγραφή Ansi C++ μαζί με μερικά χαρακτηριστικά γνωρίσματα C++0x. Η Visual C++ μπορεί επίσης να χρησιμοποιήσει τον σχεδιαστή μορφών του Visual Studio για να σχεδιάσει UI γραφικά. Επίσης υποστηρίζει τη χρήση των *εγγενών λειτουργιών*, που είναι λειτουργίες που αναγνωρίζονται από τον ίδιο το μεταγλωττιστή και που δεν εφαρμόζονται ως βιβλιοθήκη. Οι εγγενείς λειτουργίες χρησιμοποιούνται για να εκθέσουν το σύνολο οδηγίας SSE των σύγχρονων CPU

3.2.2.2 *Microsoft Visual C#*

Η Microsoft Visual C# είναι η προσθήκη της γλώσσας Microsoft C#, που στοχεύει στο .NET Framework, μαζί με τις υπηρεσίες που επιτρέπουν στο Visual Studio IDE να υποστηρίξει προγράμματα γραμμένα σε C#. Ενώ οι υπηρεσίες της γλώσσας είναι ένα μέρος του Visual Studio, ο μεταγλωττιστής είναι διαθέσιμος χωριστά ως μέρος του .NET Framework. Ο μεταγλωττιστής της Visual C# 2008 υποστηρίζει την έκδοση 3.0 των γλωσσικών C# προδιαγραφών. Η Visual C# υποστηρίζει το σχεδιαστή κλάσεων του Visual Studio, το σχεδιαστή μορφών παραθύρων, και το σχεδιαστή στοιχείων μεταξύ των άλλων.

3.2.2.3 *Microsoft Visual Basic*

Η Microsoft Visual Basic της είναι η προσθήκη της γλώσσας Microsoft VB.NET και των σχετικών εργαλείων και υπηρεσιών στο Visual Studio IDE. Εισήχθη για πρώτη φορά στο Visual Studio .NET (2002). Η Microsoft έχει τοποθετήσει τη Visual Basic για τη γρήγορη ανάπτυξη εφαρμογών (Rapid Application Development – RAD). Η Visual Basic μπορεί να χρησιμοποιηθεί τόσο στις εφαρμογές κονσολών συντακτών όσο και στις εφαρμογές με γραφικό περιβάλλον. Όπως και η Visual C#, η Visual Basic υποστηρίζει επίσης τον σχεδιαστή κλάσεων του Visual Studio, το σχεδιαστή μορφών παραθύρων, και το σχεδιαστή στοιχείων μεταξύ των άλλων. Όπως η C#, ο μεταγλωττιστής της γλώσσας VB.NET είναι επίσης διαθέσιμος ως μέρος του .NET Framework αλλά οι υπηρεσίες, που αφήνουν τα προγράμματα VB.NET να αναπτυχθούν με το Visual Studio, είναι διαθέσιμες ως μέρος των τελευταίων.

3.2.2.4 *Microsoft Visual Web Developer*

Ο Microsoft Visual Web Developer χρησιμοποιείται για να δημιουργήσει τους ιστοχώρους (Web Sites), την εφαρμογή Web και τις υπηρεσίες Web χρησιμοποιώντας

ASP.NET. Μπορούν να χρησιμοποιηθούν σαν γλώσσες προγραμματισμού και η C# αλλά και η VB.NET. Ο Visual Web Developer μπορεί να χρησιμοποιήσει τον Web Developer Visual Studio για να σχεδιάσει γραφικά τα σχεδιαγράμματα ιστοσελίδας.

3.3 Microsoft Visual Basic

3.3.1 Γενικά

Η **Visual Basic (VB)** είναι γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM. Η VB θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρησιμοποίηση, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον Χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC. Η Visual Basic προέρχεται από τη BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD) με Γραφικό Περιβάλλον Χρήστη (GUI), πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects, ή ActiveX Data Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων. Οι γλώσσες προγραμματισμού τύπου "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις. Ένας προγραμματιστής μπορεί να ολοκληρώσει μια εφαρμογή χρησιμοποιώντας τα στοιχεία που παρέχονται με την Visual Basic. Προγράμματα γραμμένα σε Visual Basic μπορούν, επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων.

Η τελική έκδοση 6 βγήκε το 1998. Η εκτεταμένη υποστήριξη της Microsoft έληξε το Μάρτιο του 2008 και ορίστηκε διάδοχος της η Visual Basic.NET (γνωστή απλά ως Visual Basic).

3.3.2 Χαρακτηριστικά της Γλώσσας

Όπως και η γλώσσα προγραμματισμού BASIC, η Visual Basic έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και το χειρισμό. Η γλώσσα δεν επιτρέπει στους προγραμματιστές να δημιουργήσουν μόνο απλές εφαρμογές GUI, αλλά μπορούν, επίσης, να αναπτύξουν πολύπλοκες εφαρμογές. Ο προγραμματισμός σε VB συνίσταται από τον οπτικό συνδυασμό στοιχείων ή ελέγχων σε μια φόρμα, τον προσδιορισμό χαρακτηριστικών και ενεργειών αυτών των στοιχείων και την σύνταξη επιπλέον γραμμών κώδικα για αυξημένη λειτουργικότητα. Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.

Αν και τα προγράμματα μπορούν να μετατραπούν σε εγγενή εκτελέσιμο κώδικα από την έκδοση 5 και μετά, αυτά εξακολουθούν να απαιτούν την παρουσία των βιβλιοθηκών χρόνου εκτέλεσης (runtime) με μέγεθος περίπου 1 MB. Οι βιβλιοθήκες runtime υπάρχουν στα Windows 2000 και αργότερα, αλλά στις παλαιότερες εκδόσεις των Windows όπως τα 95/98/NT πρέπει να διανέμονται μαζί με το εκτελέσιμο αρχείο.

Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές "σύρε κι άσε" (drag-and-drop). Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή. Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος, παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο, και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγθεί.

Εναλλακτικά, ένα συστατικό της Visual Basic μπορεί να μην έχει Γραφικό Περιβάλλον Χρήστη, αλλά, αντ' αυτού, να παρέχει αντικείμενα ActiveX σε άλλα προγράμματα μέσω Component Object Model (COM). Αυτό επιτρέπει επεξεργασία στην πλευρά του διακομιστή (server-side processing) ή τη δημιουργία πρόσθετων μορφωμάτων (add-in module).

Η γλώσσα έχει αυτόματη διαχείριση μνήμης τύπου garbage collection χρησιμοποιώντας υπολογισμό αναφορών και έχει μια μεγάλη βιβλιοθήκη με βοηθητικά αντικείμενα καθώς και βασική αντικειμενοστραφή υποστήριξη. Από τα πιο κοινά στοιχεία που περιλαμβάνονται στο προεπιλεγμένο πρότυπο έργου, ο προγραμματιστής σπάνια

χρειάζεται να καθορίσει πρόσθετες βιβλιοθήκες. Αντίθετα με πολλές άλλες γλώσσες προγραμματισμού η Visual Basic γενικά δεν διαχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες, αν και θα μετατρέψει τις λέξεις-κλειδιά σε μία τυπική διαμόρφωση. Οι συγκρίσεις συμβολοσειρών διαχωρίζουν τα πεζά από τα κεφαλαία από προεπιλογή, αλλά μπορεί να αλλάξει αυτό, εφόσον το επιθυμείτε.

Ο μεταγλωττιστής της Visual Basic είναι κοινός με τις άλλες γλώσσες του Visual Studio (C, C++), αλλά οι περιορισμοί στον IDE δεν επιτρέπουν τη δημιουργία ορισμένων στόχων (μοντέλα Windows DLL) και σε μοντέλα νημάτων.

3.4 Microsoft .NET Framework

3.4.1 Γενικά

Το .NET Framework της Microsoft είναι ένα πλαίσιο λογισμικού που μπορεί να εγκατασταθεί στους υπολογιστές που τρέχουν τα λειτουργικά συστήματα του Microsoft Windows. Περιλαμβάνει μια μεγάλη βιβλιοθήκη των κωδικοποιημένων λύσεων στα κοινά προβλήματα προγραμματισμού και μια εικονική μηχανή που διαχειρίζεται την εκτέλεση των προγραμμάτων που γράφονται συγκεκριμένα για το Framework. Το .NET Framework είναι μια προσφορά της Microsoft και προορίζεται να χρησιμοποιηθεί από τις περισσότερες νέες αιτήσεις που δημιουργούνται για την πλατφόρμα παραθύρων.

3.4.2 Base Class Library

Η BCL (Base Class Library) παρέχει μια μεγάλη σειρά των χαρακτηριστικών γνωρισμάτων των εφαρμογών συμπεριλαμβανομένων του user interface, της πρόσβασης και διαχείρισης δεδομένων, της συνδεσιμότητας των εφαρμογών με βάσεις δεδομένων, του συστήματος κρυπτογραφίας, της ανάπτυξης εφαρμογών Ιστού (web applications), των αριθμητικών αλγορίθμων, και των επικοινωνιών δικτύων. Η βιβλιοθήκη κατηγορίας χρησιμοποιείται από τους προγραμματιστές, οι οποίοι την συνδυάζουν με τον κώδικά τους για να παραγάγουν τις εφαρμογές.

3.4.3 Χρόνος Εκτέλεσης Κοινής Γλώσσας

Τα προγράμματα που γράφονται για το .NET Framework εκτελούνται σε ένα περιβάλλον λογισμικού που διαχειρίζεται τις απαιτήσεις χρόνου εκτέλεσης του προγράμματος. Επίσης μέρος του .NET Framework, αυτό το περιβάλλον χρόνου εκτέλεσης είναι γνωστό ως χρόνος εκτέλεσης κοινής γλώσσας (Common Language Runtime - CLR). Το CLR παρέχει την εμφάνιση μιας εικονικής μηχανής εφαρμογών έτσι ώστε οι προγραμματιστές δεν χρειάζονται να εξετάσουν τις ικανότητες της συγκεκριμένης CPU που θα εκτελέσει το πρόγραμμα. Το CLR παρέχει επίσης άλλες σημαντικές υπηρεσίες όπως η ασφάλεια, τη διαχείριση μνήμης, και το χειρισμό εξαίρεσης. Η βασική βιβλιοθήκη κλάσεων

και το CLR αποτελούν από κοινού το .NET Framework.

Η έκδοση 3.0 του .NET Framework συμπεριλαμβάνεται με τον Windows Server 2008 και Windows Vista. Η τρέχουσα έκδοση του Framework μπορεί επίσης να εγκατασταθεί στα Windows XP καθώς και στα Windows Server 2003. Μια μειωμένη έκδοση του .NET Framework, το .NET Compact Framework, είναι επίσης διαθέσιμη στις πλατφόρμες του Windows Mobile.

3.4.4 Κύρια Χαρακτηριστικά Σχεδιασμού

Διαλειτουργικότητα

Επειδή η αλληλεπίδραση μεταξύ των νέων και παλαιότερων εφαρμογών απαιτείται συνήθως, το .NET Framework παρέχει τα μέσα να προσεγγιστεί η λειτουργία που εφαρμόζεται στα προγράμματα που εκτελούν έξω από το .NET περιβάλλον. Η πρόσβαση στα τμήματα COM (Component Object Model) παρέχεται στο System.Runtime.InteropServices και το System.EnterpriseServices namespace του framework ενώ η πρόσβαση σε άλλη λειτουργία παρέχεται χρησιμοποιώντας το χαρακτηριστικό γνώρισμα P/Invoke.

Κοινή μηχανή χρόνου εκτέλεσης (Common Runtime Engine)

Ο χρόνος εκτέλεσης κοινής γλώσσας (CLR) είναι το τμήμα της εικονικής μηχανής του .NET framework. Όλα τα προγράμματα που χρησιμοποιούν το .NET Framework εκτελούνται κάτω από τη επίβλεψη του CLR, που εγγυάται ορισμένες ιδιότητες και συμπεριφορές στους τομείς της διαχείρισης μνήμης, της ασφάλειας, και της διαχείρισης των εξαιρέσεων.

Γλωσσική ανεξαρτησία

Το .NET Framework εισάγει ένα κοινό σύστημα τύπων (Common Type System – CTS). Η προδιαγραφή CTS καθορίζει όλους τους πιθανούς τύπους δεδομένων και τα τμήματα των προγραμμάτων που υποστηρίζονται από το CLR και πώς μπορούν ή δεν μπορούν να αλληλεπιδράσουν το ένα με το άλλο. Λόγω αυτού του χαρακτηριστικού γνωρίσματος, το .NET Framework υποστηρίζει την ανταλλαγή των περιπτώσεων τύπων μεταξύ των προγραμμάτων που γράφονται σε οποιεσδήποτε από τις .NET γλώσσες.

Base Class Library

Η BCL (Base Class Library) αποτελεί μέρος της βιβλιοθήκης κλάσεων του framework (Framework Class Library - FCL), είναι μια βιβλιοθήκη διαθέσιμη σε όλες τις γλώσσες που χρησιμοποιούν το .NET Framework. Το BCL παρέχει τις κατηγορίες που τοποθετούν διάφορες κοινές λειτουργίες, συμπεριλαμβανομένης της ανάγνωσης και του γραψίματος αρχείων, της γραφικής απόδοσης, της αλληλεπίδρασης βάσεων

δεδομένων και του χειρισμού εγγράφων σε XML.

Απλουστευμένη επέκταση

Το .NET Framework περιλαμβάνει τα χαρακτηριστικά σχεδιασμού και τα εργαλεία που βοηθούν να κατορθώσουν την εγκατάσταση ενός λογισμικού εξασφαλίζοντας ότι δεν παρεμποδίζει το προηγούμενος εγκατεστημένο λογισμικό, και ότι προσαρμόζεται στις απαιτήσεις ασφάλειας.

Φορητότητα

Η σχεδίαση του .NET framework επιτρέπει σε ένα πρόγραμμα που χρησιμοποιεί το Framework να τρέξει χωρίς καμμία αλλαγή σε οποιοδήποτε τύπο συστήματος. Αυτό επιτρέπει σε τρίτους να δημιουργήσουν εφαρμογές συμβατές με το framework και των γλωσσών του σε άλλες πλατφόρμες.

3.4.5 .NET Framework 3.5

Η έκδοση 3.5 του .NET Framework κυκλοφόρησε στις 19 Νοεμβρίου 2007, αλλά δεν συμπεριλαμβάνεται με τον Windows Server 2008. Η έκδοση 3.5 του .NET Framework χρησιμοποιεί το CLR της έκδοσης 2.0. Επιπλέον, εγκαθιστά το .NET Framework 2.0 SP1 και το .NET Framework 3.0 SP1, το οποίο προσθέτει μερικές μεθόδους και ιδιότητες στις κατηγορίες BCL στην έκδοση 2.0 που απαιτούνται για τα χαρακτηριστικά γνωρίσματα έκδοσης 3.5 όπως η ενσωματωμένη γλώσσα ερωτημάτων (Language Integrated Query - LINQ). Αυτές οι αλλαγές δεν έχουν επιπτώσεις στις εφαρμογές που γράφονται για την έκδοση 2.0.

Όπως με τις προηγούμενες εκδόσεις, ένα νέο .NET Compact Framework 3.5 καταβλήθηκε με αυτήν την αναπροσαρμογή προκειμένου να παρασχεθεί η υποστήριξη για τα πρόσθετα χαρακτηριστικά γνωρίσματα στο Windows Mobile και τις Windows Embedded CE συσκευές.

3.4.5.1 Αλλαγές από την έκδοση 3.0

- Νέα γλωσσικά χαρακτηριστικά γνωρίσματα σε C# 3.0 και το μεταγλωττιστή VB.NET 9.0
- Μέθοδοι επέκτασης
- Ανώνυμοι τύποι με το στατικό συμπέρασμα τύπων
- Ενσωματωμένη γλώσσα ερωτημάτων (Language Integrated Query - LINQ) μαζί με τους διάφορους προμηθευτές του
- LINQ στα αντικείμενα
- LINQ σε XML
- LINQ στο SQL
- Υποστήριξη σελιδοποίησης για ADO.NET

- ADO.NET συγχρονισμός API για να συγχρονίσει τις τοπικές caches και την αποθήκευση δεδομένων από την πλευρά των servers.
- Ασύγχρονο δίκτυο I/O API.
- Όμοιος σωρός δικτύωσης, συμπεριλαμβανομένου ενός διοικούμενου PNRP resolver.
- Συμπεριλαμβάνεται η ASP.NET AJAX
- Νέο System.CodeDom namespace.

Κεφάλαιο 4

4.1 Γενικά

Κατά την Μελέτη μιας διαδικτυακής εφαρμογής είναι προφανές ότι θα χρειαστεί να δημιουργηθούν κάποιες βασικές λειτουργίες για τη διαχείριση κάποιων πιθανών επαναλαμβανόμενων γεγονότων καθώς και την επίλυση κάποιων κομβικών λειτουργικών προβλημάτων. Στην παρούσα εργασία, για παράδειγμα, είναι οφθαλμοφανές ότι θα χρειαστούμε μια κλάση η οποία θα είναι υπεύθυνη για την σύνδεση με τον απομακρυσμένο διακομιστή (server) καθώς και για την αποστολή των κατάλληλων εντολών για την επίτευξη του επιθυμητού αποτελέσματος.

4.2 Βασικές Λειτουργίες της Εφαρμογής

Σύνδεση με τον Απομακρυσμένο Διακομιστή (Server)

Για την σύνδεση με τον απομακρυσμένο διακομιστή (server) χρησιμοποιήθηκε η κλάση `FtpWebRequest`, η οποία παίρνει σαν όρισμα την FTP διεύθυνση που επιθυμούμε να συνδεθούμε. Η κλάση `FtpWebRequest` αποτελεί μέρος του .Net Framework και πιο συγκεκριμένα του System.Net. Με την μέθοδο `FtpWebRequest.Create` κάνουμε αίτημα στον σερβερ για την δημιουργία μιας σύνδεσης. Επόμενο βήμα είναι να στείλουμε στον σερβερ, αν υπάρχουν, το Username και το Password με την μέθοδο `FtpWebRequest.Credentials`.

Αποστολή Αιτημάτων στον Απομακρυσμένο Διακομιστή (Server)

Μετά την επιτυχημένη εγκαθίδρυση μιας συνδεσης με τον σερβερ, επόμενο βήμα είναι η αποστολή των διαφόρων αιτημάτων μας. Αυτό επιτυγχάνεται με τη χρήση της μεθόδου `FtpWebRequest.Method`, με την οποία στέλνουμε την εντολή που επιθυμούμε στον σερβερ. Ένα από τα βασικά μειονεκτήματα της `FtpWebRequest` είναι ότι περιορίζει το χρήστη όσον αφορά τις εντολές που μπορεί να στείλει στον FTP Server. Έτσι ελλείψεις βασικών εντολών όπως η CD και η CDUP καθιστούν αδύνατη την πλοήγηση μέσα στον FTP Server, καθώς και η έλλειψη των εντολών MLST και MLSL δυσκολεύουν το έργο μας για ένα σωστό Directory Listing.

Λήψη Αποκρίσεων του Απομακρυσμένου Διακομιστή (Server)

Μετά την αποστολή ενός αιτήματος στον σερβερ χρησιμοποιούμε την κλάση `FtpWebResponse` για την λήψη της απάντησης στο αίτημα που κάναμε, ορίζοντας ως τιμή της `FtpWebResponse` το αποτέλεσμα της μεθόδου

FtpWebRequest.GetResponse. Σε περίπτωση που η απάντηση στο αίτημά μας αποτελείται από μια ροή δεδομένων (stream), τότε πρέπει να δημιουργήσουμε μια μεταβλητή τύπου stream στην οποία θα αποθηκεύονται προσωρινά τα δεδομένα της απάντησής μας.

Μέθοδος Δημιουργίας Περιεχομένων Φακέλου (Indexer)

Η μέθοδος δημιουργίας περιεχομένων φακέλου δέχεται ως όρισμα τη διαδρομή ενός φακέλου στον τοπικό υπολογιστή (Root Directory) και δημιουργεί δυο λίστες με όλους τους υποφακέλους και τα αρχεία που βρίσκονται μέσα στον Root Directory. Η μέθοδος προσθέτει, για κάθε φάκελο και υποφάκελο που βρίσκονται μέσα στον Root Directory, ένα στοιχείο τύπου DirectoryInfo στην πρώτη λίστα και ένα στοιχείο τύπου FileInfo για κάθε αρχείο που υπάρχει μέσα στον Root Directory. Τα στοιχεία τύπων DirectoryInfo και FileInfo είναι μεταβλητές του συστήματος οι οποίες περιέχουν όλες τις απαραίτητες πληροφορίες του συστήματος για τον εκάστοτε φάκελο ή αρχείο, όπως ημερομηνία και ώρα δημιουργίας, ημερομηνία και ώρα τελευταίας τροποποίησης κ.α.

4.3 Συστατικά της εφαρμογής

Το πρόγραμμα μας αποτελείται από τις κλάσεις LoginForm.vb, DownloadForm.vb και WidgetForm.vb, οι οποίες είναι τύπου WindowsForm και αποτελούν όλες τις γραφικές απεικονίσεις (Window) της εφαρμογής, καθώς και από την κλάση CommonClasses.vb, η οποία είναι τύπου Class και περιέχει όλες τις κλάσεις που χρησιμοποιούν από κοινού οι φόρμες της εφαρμογής.

Κάθε μια από τις προαναφερθέντες κλάσεις αποτελεί μια ξεχωριστή οντότητα και είναι υπεύθυνη για την εκτέλεση συγκεκριμένων λειτουργιών. Πιο αναλυτικά:

4.3.1 Η Κλάση CommonClasses

Η κλάση CommonClasses.vb περιέχει όλες τις κλάσεις που χρησιμοποιούνται από κοινού από τις φόρμες της εφαρμογής. Αποτελείται από τις κλάσεις FtpOperations και Indexer και από το module PublicVariables.

4.3.1.1 Η κλάση FtpOperations

Η κλάση FtpOperations είναι υπεύθυνη για την σύνδεση με τον FTP Server καθώς και για την αποστολή των αιτημάτων του χρήστη. Αποτελείται από τον Constructor και τις μεθόδους διαχείρισης αιτημάτων στον FTP Server. Πιο συγκεκριμένα:

Ο Constructor της κλάσης FtpOperations

```
Public Sub New(ByVal Host As String, ByVal Username As String, ByVal Password As String)
```

Ο Constructor της κλάσης FtpOperations παίρνει σαν ορίσματα την διεύθυνση της FTP περιοχής με την οποία επιθυμούμε να συνδεθούμε (Host), καθώς και τα πιστοποιητικά της σύνδεσης (Username,Password) και τα αποθηκεύει στις Private μεταβλητές FTPHost,FTPUsername και FTPPassword αντίστοιχα.

Οι μέθοδοι της κλάσης FtpOperations

Οι μέθοδοι της κλάσης FtpOperations που είναι υπεύθυνες για την επικοινωνία με τον FTP Server, χρησιμοποιούν τις μεθόδους FtpWebRequest για τη σύνδεση και την αποστολή των αιτημάτων και την FtpWebResponse για την λήψη των αποκρίσεων του FTP Server.

Public Function DownloadFile(ByVal DownloadFileAbsolutePath As String) As Boolean

Η μέθοδος DownloadFile δέχεται σαν όρισμα το Absolute Path του αρχείου, δηλαδή την διαδρομή του αρχείου μέσα στον Root Folder, και επιστρέφει μια τιμή τύπου Boolean μετά την επιτυχημένη ή όχι ολοκλήρωση της διαδικασίας. Η μέθοδος χρησιμοποιεί μια μεταβλητή τύπου Stream για την προσωρινή αποθήκευση της ροή δεδομένων (Data Stream) της απόκρισης του FTP Server (FtpWebResponse.GetResponseStream) και έπειτα χρησιμοποιώντας μια μεταβλητή τύπου FileStream αποθηκεύει το αρχείο στον φάκελο του χρήστη στον τοπικό υπολογιστή.

Public Sub UploadFile(ByVal UploadFileFullPath As String)

Η μέθοδος DownloadFile δέχεται σαν όρισμα το μονοπάτι του αρχείου στον τοπικό υπολογιστή και χρησιμοποιώντας το Absolute Path του αρχείου (την διαδρομή του μέσα στον Root Folder) δημιουργεί μια σύνδεση με τον FTP Server και στέλνει το αίτημα για ανέβασμα (Upload) του αρχείου . Η μέθοδος χρησιμοποιεί μια μεταβλητή τύπου FileStream για να μετατρέψει τα δεδομένα του αρχείου σε ροή δεδομένων (Data Stream) και έπειτα χρησιμοποιώντας μια μεταβλητή τύπου Stream, αποστέλλει τα δεδομένα στον FTP Server.

Public Sub DeleteFile(ByVal DeleteFileFullPath As String)

Η μέθοδος DeleteFile δέχεται σαν όρισμα το μονοπάτι του αρχείου στον τοπικό υπολογιστή και χρησιμοποιώντας το Absolute Path του αρχείου (την διαδρομή του μέσα στον Root Folder) δημιουργεί μια σύνδεση με τον FTP Server και στέλνει το αίτημα για διαγραφή (Delete) του αρχείου.

Public Function MakeDirectory(ByVal MakeDirectoryFullPath As String) As Boolean

Η μέθοδος MakeDirectory δέχεται σαν όρισμα το μονοπάτι του φακέλου στον τοπικό υπολογιστή και χρησιμοποιώντας το Absolute Path του (την διαδρομή του φακέλου μέσα

στον Root Folder) δημιουργεί μια σύνδεση με τον FTP Server και στέλνει το αίτημα για δημιουργία (MakeDirectory) του φακέλου

Public Function RemoveDirectory(ByVal RemoveDirectoryFullPath As String) As Boolean

Η μέθοδος RemoveDirectory δέχεται σαν όρισμα το μονοπάτι του φακέλου στον τοπικό υπολογιστή και χρησιμοποιώντας το Absolute Path του (την διαδρομή του φακέλου μέσα στον Root Folder) δημιουργεί μια σύνδεση με τον FTP Server και στέλνει το αίτημα για διαγραφή (Delete) του φακέλου.

Εκτός από τις μεθόδους που αφορούν την διαχείριση της σύνδεσης, αποστολής αιτημάτων και λήψης απαντήσεων του FTP Server, η κλάση FtpOperations περιέχει και μεθόδους για τον έλεγχο της μορφοποίησης (format) διάφορων μεταβλητών. Πιο συγκεκριμένα:

Private Function FormatHost(ByVal CurrentFTPHost As String) As String

Η μέθοδος FormatHost δέχεται σαν όρισμα (τύπου String) την διεύθυνση ενός FTP Server, ελέγχει αν είναι της μορφής <ftp://XXXXXX.XXX>, κάνει, αν χρειάζεται, τις απαραίτητες τροποποιήσεις και επιστρέφει την επιθυμητή μορφή σαν μεταβλητή τύπου String.

Private Function FormatFileHost(ByVal AbsolutePath As String) As String

Η μέθοδος FormatHost δέχεται σαν όρισμα (τύπου String) το Absolute Path ενός αρχείου ή φακέλου και το συνδυάζει με τη διεύθυνση του FTP Server, επιστρέφοντας σαν μεταβλητή τύπου String τη διεύθυνση ενός αρχείου ή φακέλου στον FTP Server.

Private Function GetAbsolutePath(ByVal FullPath As String) As String

Η μέθοδος GetAbsolutePath δέχεται σαν όρισμα (τύπου String) τη διαδρομή ενός αρχείου ή φακέλου που βρίσκεται μέσα στον φάκελο του χρήστη στον τοπικό υπολογιστή και επιστρέφει σαν μεταβλητή τύπου String το Absolute Path (την διαδρομή του αρχείου ή φακέλου μέσα στον Root Folder).

Τέλος, η FtpOperations περιέχει τις κλάσεις GetFtpHost(), GetUsername() και GetPassword, οι οποίες επιστρέφουν τις τιμές των Private μεταβλητών που περιέχουν τη διεύθυνση του FTP Server, το username και το password αντίστοιχα.

4.3.1.2 Η Κλάση Indexer

Η κλάση Indexer είναι υπεύθυνη για την δημιουργία δυο λιστών με τα περιεχόμενα ενός φακέλου. Η πρώτη λίστα περιέχει όλους τους υποφακέλους ενώ η δεύτερη όλα τα αρχεία που υπάρχουν στον φάκελο που επιθυμούμε. Η κλάση αποτελείται από τον Constructor και τις μεθόδους δημιουργίας και διαχείρισης των λιστών. Πιο συγκεκριμένα:

Ο Constructor της κλάσης Indexer

```
Public Sub New(ByVal DirectoryPath As String)
```

Ο Constructor της κλάσης Indexer δέχεται σαν ορίσμα (τύπου String) το μονοπάτι του φακέλου που επιθυμούμε να δημιουργήσουμε τις λίστες περιεχομένων, το αποθηκεύει σε μια Private μεταβλητή και καλεί τη μέθοδο GetIndexer η οποία δημιουργεί τις λίστες περιεχομένων όπως θα εξηγήσουμε παρακάτω.

Οι μέθοδοι της κλάσης Indexer

```
Private Sub GetIndexer(ByVal CurrentDirectoryPath As String)
```

Η μέθοδος GetIndexer δέχεται σαν ορίσμα (τύπου String) το μονοπάτι του φακέλου που επιθυμούμε να δημιουργήσουμε λίστες περιεχομένων και δημιουργεί μια μεταβλητή τύπου DirectoryInfo η οποία περιέχει όλες τις πληροφορίες του φακέλου. Χρησιμοποιώντας τις μεθόδους DirectoryInfo.GetDirectories, η οποία επιστρέφει μία λίστα τύπου DirectoryInfo με όλους τους υποφακέλους που υπάρχουν μέσα στο φάκελο που καλεί τη μέθοδο, και DirectoryInfo.GetFiles, η οποία επιστρέφει μία λίστα τύπου FileInfo με όλα τα αρχεία που υπάρχουν μέσα στο φάκελο που καλεί τη μέθοδο, η μέθοδος δημιουργεί τις δύο λίστες με τα περιεχόμενα του φακέλου. Η μέθοδος καλεί τον εαυτό της για κάθε στοιχείο της λίστας που επιστρέφει η DirectoryInfo.GetDirectories προσθέτοντας στις λίστες περιεχομένων όλα τα στοιχεία των υποφακέλων του φακέλου που έγινε το indexing.

```
Private Sub GetIndexer(ByVal CurrentDirectoryPath As String)
```

```
    Dim DirectoryInfo As New DirectoryInfo(CurrentDirectoryPath)
```

```
    For Each SubDirectoryInfo In DirectoryInfo.GetDirectories()
```

```
        DirectoryList.Add(SubDirectoryInfo)
```

```
        GetIndexer(SubDirectoryInfo.FullName)
```

```
    Next
```

```
    For Each fi In DirectoryInfo.GetFiles()
```

```
        FileList.Add(fi)
```

Next

End Sub

Public Function GetFiles() **As** List(Of IO.FileInfo)

Η μέθοδος GetFiles επιστρέφει την λίστα με τα αρχεία που δημιούργησε η μέθοδος GetIndexer.

Public Function GetDirectories() **As** List(Of IO.DirectoryInfo)

Η μέθοδος GetDirectories επιστρέφει την λίστα με τους φακέλους που δημιούργησε η μέθοδος GetIndexer.

Public Sub CreateIndexer()

Η μέθοδος CreateIndexer δημιουργεί ένα αρχείο τύπου NFO με τα περιεχόμενα των δυο λιστών που δημιούργησε η μέθοδος GetIndexer και το αποθηκεύει στον φάκελο στον οποίο έγινε τον indexing.

Private Function GetAbsolutePath(ByVal FullPath **As** String) **As** String

Η μέθοδος GetAbsolutePath δέχεται σαν όρισμα (τύπου String) τη διαδρομή ενός αρχείου ή φακέλου που βρίσκεται μέσα στον φάκελο του χρήστη στον τοπικό υπολογιστή και επιστρέφει σαν μεταβλητή τύπου String το Absolute Path (την διαδρομή του αρχείου ή φακέλου μέσα στον Root Folder).

4.3.1.2 To Module PublicVariables

Το module PublicVariables περιέχει τις μεταβλητές που χρησιμοποιούνται από όλες τις κλάσεις και μεθόδους της εφαρμογής. Το module, αντίθετα από την κλάση, δεν μπορεί να γίνει inherited ή implemented από μια κλάση ενώ όλα τα στοιχεία του είναι shared.

Public Function GetUserFolder(ByVal Username **As** String) **As** String

Η μέθοδος GetUserFolder δέχεται σαν όρισμα (τύπου String) το username μιας FtpOperations κλάσης και επιστρέφει το μονοπάτι για τον φάκελο του χρήστη που δημιουργεί η εφαρμογή στην επιφάνεια εργασίας του υπολογιστή.

Public Function ReadIndexLog() **As** List(Of String)

Η μέθοδος ReadIndexLog δημιουργεί και επιστρέφει μια λίστα τύπου String με τα περιεχόμενα του αρχείου IndexLog.nfo που δημιουργεί η μέθοδος CreateIndexer.

Public Function IndexLogFullPathList(ByVal Username **As** String) **As** List(Of String)

Η μέθοδος `IndexLogFullPathList`, χρησιμοποιώντας τη λίστα που επιστρέφει η μέθοδος `ReadIndexLog`, δημιουργεί και επιστρέφει μια λίστα τύπου `String` με τα περιεχόμενα του αρχείου `IndexLog.nfo` συνδυασμένα, χρησιμοποιώντας τη μέθοδο `Combine` της κλάσης `Path` του `System.IO`, με το μονοπάτι του φακέλου του χρήστη στον τοπικό υπολογιστή.

4.3.2 LoginForm

4.3.2.1 Γενικά

Η κλάση `LoginForm` η πρώτη από τις τρεις κλάσεις τύπου `WindowsForm` που χρησιμοποιεί η εφαρμογή, οι οποίες διαδέχονται η μια την άλλη, και είναι υπεύθυνη για τον έλεγχο των μεταβλητών που δίνει ο χρήστης (διεύθυνση FTP, username και password), το κατέβασμα του αρχείου `IndexLog.nfo` από τον FTP Server και την δημιουργία και απόκρυψη του φακέλου του χρήστη στην επιφάνεια εργασίας.

4.3.2.2 Γραφική Απεικόνιση

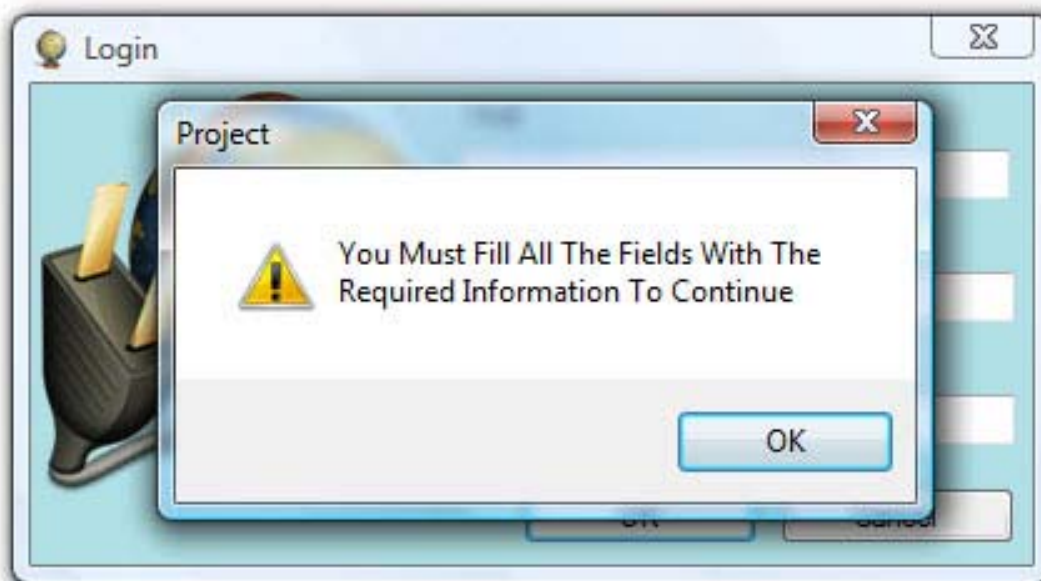
Η κλάση γραφικά αποτελείται από τρία πεδία τύπου `Textbox` (`FTPHostTextBox`, `UsernameTextBox` και `PasswordTextBox`) στα οποία συμπληρώνει ο χρήστης την διεύθυνση του FTP Server, το username και το password αντίστοιχα και δύο κουμπιά (το `OKButton` και το `CancelButton`).



Εικόνα 4.1 Η γραφική απεικόνιση της κλάσης `LoginForm`

Μετά το πάτημα του κουμπιού `OK`, η κλάση καλεί την μέθοδο `CheckForEmptyFields` για να ελέγξει αν τα όλα πεδία στη φόρμα (`FTPHostTextBox`, `UsernameTextBox` και

PasswordTextBox) είναι συμπληρωμένα. Σε περίπτωση που υπάρχει άδειο πεδίο η μέθοδος ενημερώνει τον χρήστη και επιστρέφει στην αρχική κατάσταση.



Εικόνα 4.2 Προειδοποίηση συστήματος σε περίπτωση ύπαρξης άδειων πεδίων

Σε περίπτωση που δεν υπάρχει άδειο πεδίο η κλάση καλεί τη μέθοδο CheckFTPConnection η οποία ελέγχει την σύνδεση βάσει των στοιχείων που συμπλήρωσε ο χρήστης στη φόρμα χρησιμοποιώντας τη μέθοδο DownloadFile της κλάσης FtpOperations για να κατεβάσει το αρχείο IndexLog.nfo από τον FTP Server στο φάκελο Temp του συστήματος. Αν το κατέβασμα ολοκληρωθεί χωρίς κανένα πρόβλημα, η μέθοδος δημιουργεί τον φάκελο του χρήστη στην επιφάνεια εργασίας του υπολογιστή και τον αποκρύπτει από το σύστημα. Αμέσως μετά καλεί την κλάση DownloadForm της εφαρμογής.

4.3.2.3 Οι Μέθοδοι Της κλάσης LoginForm

Private Function CheckForEmptyFields() As Boolean

Η μέθοδος CheckForEmptyFields ελέγχει αν τα όλα πεδία στη φόρμα (FTPHostTextBox, UsernameTextBox και PasswordTextBox) είναι συμπληρωμένα. Σε περίπτωση που υπάρχει άδειο πεδίο η μέθοδος επιστρέφει False, αλλιώς επιστρέφει True.

Private Function CheckFTPConnection() As Boolean

Η μέθοδος CheckFTPConnection είναι υπεύθυνη για τον έλεγχο των στοιχείων που συμπλήρωσε ο χρήστης στη φόρμα καθώς και για την δημιουργία και απόκρυψη του φακέλου του χρήστη στην επιφάνεια εργασίας. Για τον έλεγχο των στοιχείων που συμπλήρωσε ο χρήστης στη φόρμα η μέθοδος χρησιμοποιεί τη μέθοδο DownloadFile της κλάσης

FtpOperations για να κατεβάσει το αρχείο IndexLog.nfo από τον FTP Server στο φάκελο Temp του συστήματος. Αν το κατέβασμα ολοκληρωθεί χωρίς κανένα πρόβλημα, η μέθοδος δημιουργεί τον φάκελο του χρήστη στην επιφάνεια εργασίας του υπολογιστή και τον αποκρύπτει από το σύστημα.

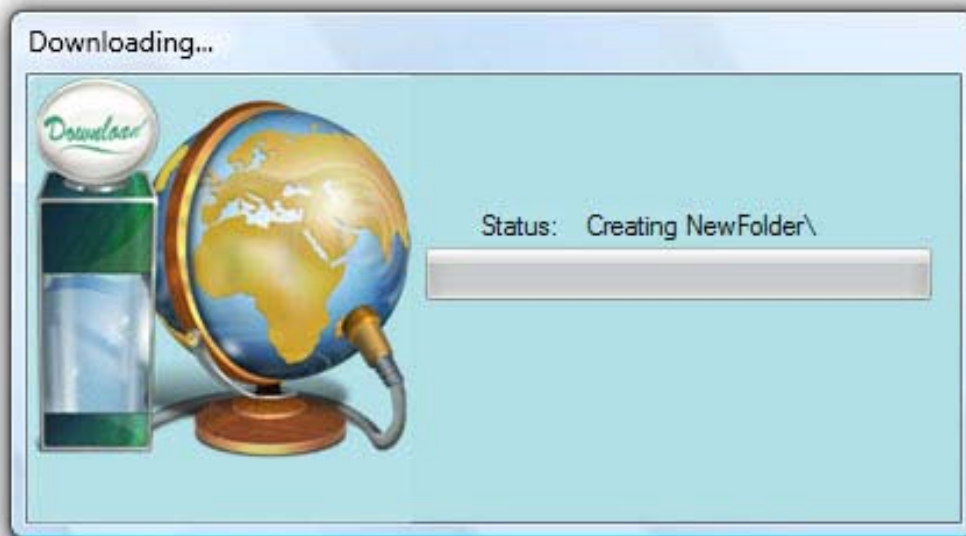
4.3.3 DownloadForm

4.3.3.1 Γενικά

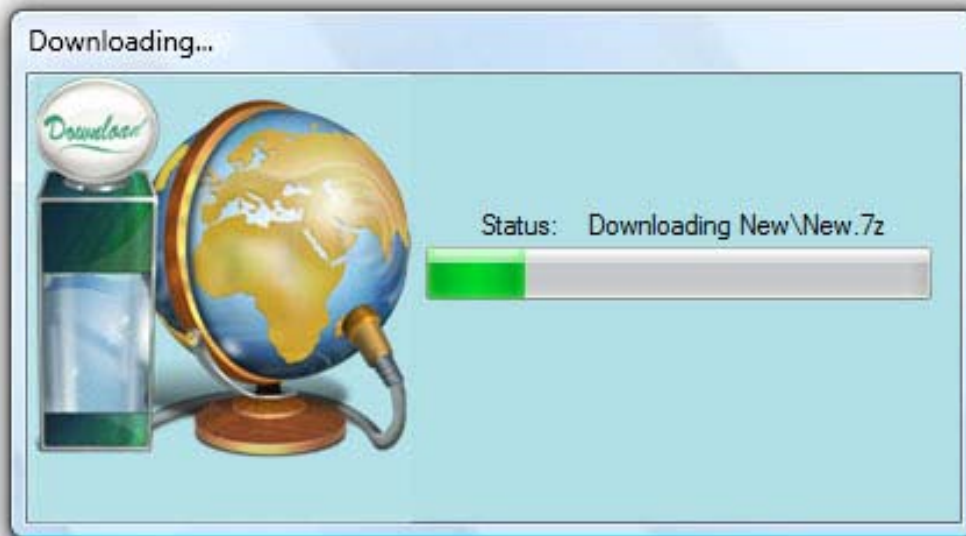
Η κλάση DownloadForm είναι υπεύθυνη για το κατέβασμα όλων των αρχείων και την δημιουργία στον τοπικό υπολογιστή όλων των φακέλων που υπάρχουν στον FTP Server.

4.3.3.2 Γραφική Απεικόνιση

Η κλάση γραφικά αποτελείται από μια μπάρα προόδου (Progress Bar), η οποία απεικονίζει γραφικά την κατάσταση της προόδου της λειτουργίας της κλάσης, και από μια ετικέτα τύπου Label στην οποία αναγράφεται το αρχείο που κατεβαίνει ή του φακέλου που δημιουργήθηκε.



Εικόνα 4.3 Δημιουργία φακέλου της κλάσης DownloadForm



Εικόνα 4.4 Κατέβασμα αρχείου της κλάσης DownloadForm

Η κλάση DownloadForm κατά την εκκίνηση καλεί τον constructor της FtpOperations χρησιμοποιώντας σαν ορίσματα τις τιμές που έδωσε ο χρήστης στα πεδία της φόρμας της κλάσης LoginForm, δημιουργώντας έτσι μια μεταβλητή τύπου FtpOperations. Έπειτα ελέγχει αν υπάρχει το αρχείο IndexLog.nfo στον φάκελο Temp του συστήματος. Αν υπάρχει τότε η κλάση κάνοντας χρήση της μεθόδου DownloadFilesFromIndexer, η οποία χρησιμοποιεί τη μέθοδο ReadIndexLog του module PublicVariables για να διαβάσει τα περιεχόμενα του αρχείου IndexLog.nfo, δημιουργεί όλους τους φακέλους και κατεβάζει όλα τα αρχεία που βρίσκονται στον FTP Server στον φάκελο του χρήστη που βρίσκεται στην επιφάνεια εργασίας του τοπικού υπολογιστή. Επίσης η μέθοδος DownloadFilesFromIndexer χρησιμοποιώντας τη μέθοδο ProgressBar αυξάνει τη μπάρα προόδου της κλάσης δείχνοντας έτσι γραφικά στον χρήστη το σημείο στο οποίο βρίσκεται πριν την ολοκλήρωση της.

4.3.3.3 Οι μέθοδοι της κλάσης DownloadForm

Private Sub DownloadFilesFromIndexer()

Η μέθοδος DownloadFilesFromIndexer είναι υπεύθυνη για το κατέβασμα όλων των αρχείων και την δημιουργία όλων των φακέλων που υπάρχουν στον FTP Server στον φάκελο του χρήστη που βρίσκεται στην επιφάνεια εργασίας του τοπικού υπολογιστή. Η μέθοδος διαβάζει τα περιεχόμενα του αρχείου IndexLog.nfo, χρησιμοποιώντας τη μέθοδο ReadIndexLog του module PublicVariables, δημιουργεί μια λίστα με τα περιεχόμενα του FTP Server που έχουν καταγραφεί στο αρχείο IndexLog.nfo και ελέγχει κάθε στοιχείο της λίστας ξεχωριστά. Αν το στοιχείο της λίστας αναφέρεται σε φάκελο τότε τον δημιουργεί στον φάκελο του χρήστη χρησιμοποιώντας τη μέθοδο CreateDirectory της κλάσης Directory που βρίσκεται στη βιβλιοθήκη System.IO. Αν το στοιχείο της λίστας αναφέρεται σε αρχείο τότε

χρησιμοποιώντας τη μέθοδο DownloadFile της κλάσης FtpOperations κατεβάζει το αρχείο στον φάκελο του χρήστη. Κατα τη διάρκεια του ελέγχου η μέθοδος για κάθε στοιχείο της λίστας αλλάζει το περιεχόμενο της ετικέτας ShowStatusLabel ανάλογα με την ενέργεια που εκτελείται ("Creating" Folder,"Downloading"File) και αυξάνει τη μπάρα προόδου χρησιμοποιώντας τη μέθοδο IncreaseProgressBar.

```
Private Sub DownloadFilesFromIndexer()  
  
    Dim FileList As List(Of String) = PublicVariables.ReadIndexLog()  
  
    For Each IndexLogItem In FileList  
  
        If IndexLogItem.EndsWith("\") Then  
  
            Directory.CreateDirectory(Path.Combine(PublicVariables.GetUserFolder( _  
                                                    FtpConnection.GetUsername), IndexLogItem))  
  
            ShowStatusLabel.Text = "Creating " & IndexLogItem  
  
            IncreaseProgressBar(FileList.Count)  
  
            Me.Refresh()  
  
            System.Threading.Thread.Sleep(300)  
  
        Else  
  
            ShowStatusLabel.Text = "Downloading " & IndexLogItem  
  
            IncreaseProgressBar(FileList.Count)  
  
            Me.Refresh()  
  
            System.Threading.Thread.Sleep(300)  
  
            FtpConnection.DownloadFile(IndexLogItem)
```


End If

Next

ShowStatusLabel.Text = "All Done!!!"

System.Threading.Thread.Sleep(200)

End Sub

Private Sub IncreaseProgressBar(**ByVal** ProgressBarStep **As Integer**)

Η μέθοδος IncreaseProgressBar είναι υπεύθυνη για την αύξηση της τιμής της μπάρας προόδου. Δέχεται σαν όρισμα έναν ακέραιο αριθμό και βάσει αυτού καθορίζεται το βήμα της αύξησης. Στην εφαρμογή μας το όρισμα της μεταβλητής ισούται με τον αριθμό των περιεχομένων του αρχείου IndexLog.nfo, δηλαδή τον αριθμό των αρχείων και φακέλων που βρίσκονται στον FTP Server.

4.3.4 WidgetForm

4.3.4.1 Γενικά

Η κλάση WidgetForm είναι υπεύθυνη τον εντοπισμό οποιασδήποτε αλλαγής εχει συμβεί σε φακέλους ή αρχεία του φακέλου του χρήστη καθώς και για την ενημέρωση του FTP Server για τις αλλαγές αυτές κατά τον τερματισμό της ή όποτε της ζητηθεί.

Κατα την εκκίνηση της η κλάση δημιουργεί μια μεταβλητή τύπου FtpOperations, η οποία παίρνει την τιμή της αντίστοιχης μεταβλητής της κλάσης DownloadForm. Επίσης δημιουργείται μια λίστα με τα περιεχόμενα του φακέλου του χρήστη, χρησιμοποιώντας τη μέθοδο GetIndexer της κλάσης Indexer. Κατα τον τερματισμό της κλάσης ή όταν ο χρήστης επιλέξει σαν ενέργεια την ενημέρωση του Server (επιλογή Update Server απο το μενού της κλάσης), δημιουργείται μια δεύτερη λίστα περιεχομένων με τα αντικείμενα που υπάρχουν εκείνη τη στιγμή στον φάκελο του χρήστη και χρησιμοποιώντας τις μεθόδους CreateToFTP και DeleteFromFTP γίνεται μια σύγκριση των δυο λιστών περιεχομένων (πρίν και μετά) και αποστέλλονται στον Server οι εντολές για τροποποιήσεις μόνο των αρχείων ή φακέλων που τροποποιήθηκαν στον τοπικό υπολογιστή.

4.3.4.2 Γραφική Απεικόνιση

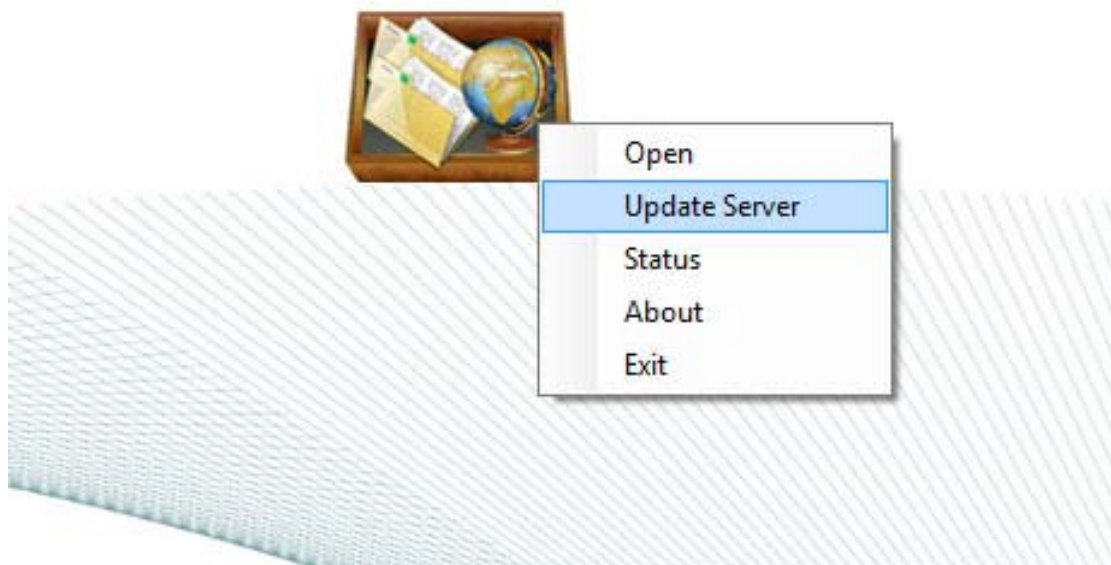
Η κλάση γραφικά αποτελείται απο ένα εικονίδιο το οποίο καλύπτει ολόκληρη τη φόρμα ενώ δεν υπάρχουν καθόλου περιθώρια παραθύρου (Border) καθώς και ελεγκτές

παραθύρου (Minimize,Maximize,Close). Η τροποποίηση του μεγέθους της κλάσης απο τον χρήστη δεν είναι δυνατή ενώ με το πάτημα του δεξιού πλήκτρου του ποντικιού ανοίγει ενα μενού με όλες τις επιλογές που είναι διαθέσιμες στο χρήστη.Η κλάση υποστηρίζει τη λειτουργία Drag And Drop, με τα αντικείμενα που γίνονται Drop να αντιγράφονται στον φάκελο του χρήστη.



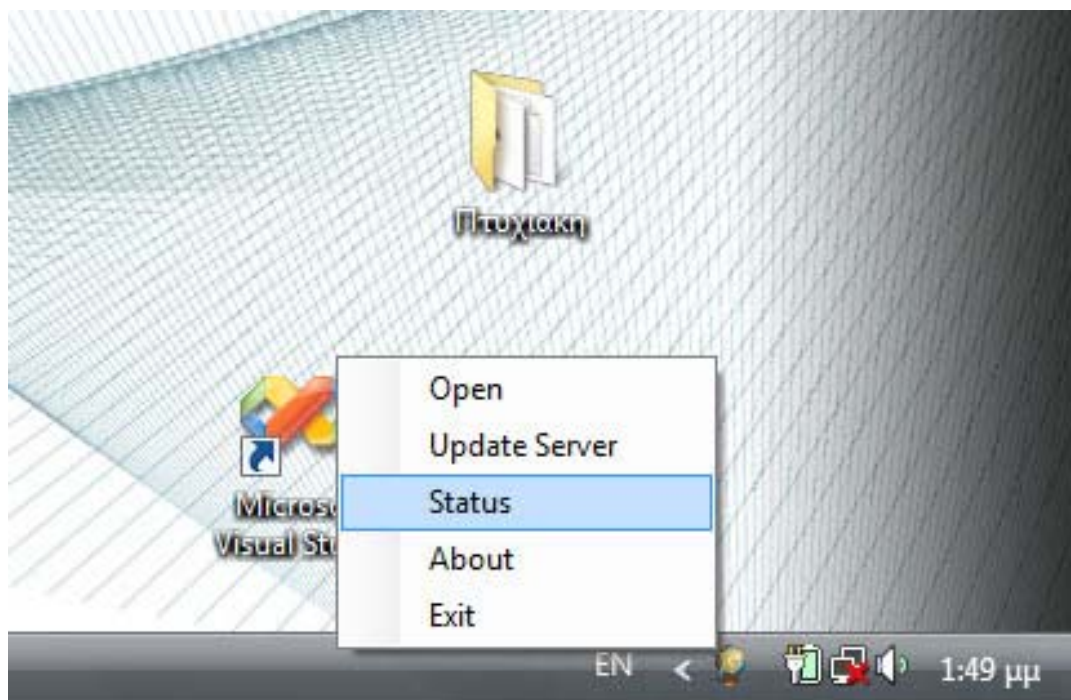
Εικόνα 4.5 Η γραφική απεικόνιση της κλάσης LoginForm

Επίσης υπάρχει ένα μενού επιλογών τύπου ContextMenuStrip το οποίο εμφανίζεται μετά το πάτημα του δεξιού πλήκτρου του ποντικιού μέσα στη κλάση.



Εικόνα 4.6 Το μενού επιλογών της κλάσης LoginForm

Επίσης δημιουργείτε ένα εικονίδιο στο System Tray το οποίο κάνει διαθέσιμο ανα πάσα στιγμή, με το πάτημα του δεξιού πλήκτρου του ποντικιού πάνω στο εικονίδιο, το μενού επιλογών του χρήστη.



Εικόνα 4.7 Το μενού επιλογών του χρήστη του εικονιδίου του System Tray

4.3.4.3 Οι λειτουργίες της κλάσης *WidgetForm*

Οι λειτουργίες της κλάσης *WidgetForm* χωρίζονται σε τρεις βασικές κατηγορίες.

WidgetForm Move Functions

Οι μέθοδοι αυτής της κατηγορίας ασχολούνται αποκλειστικά με την μετακίνηση της κλάσης μέσα στο περιβάλλον του λειτουργικού συστήματος. Η μη ύπαρξη περιθωρίων παραθύρου (*Border*) καθιστούσε μη εφικτή την μετακίνηση της φόρμας σε ένα άλλο σημείο της οθόνης. Η κλάση, με το πάτημα του αριστερού πλήκτρου του ποντικιού μέσα στη φόρμα (*Mouse Down*), αποθηκεύει προσωρινά τις συντεταγμένες του σημείου που βρίσκεται ο δείκτης του ποντικιού μέσα στη κλάση σε μια μεταβλητή τύπου *Point* που βρίσκεται στην κλάση *Drawing*. Με την απελευθέρωση του πλήκτρου (*Mouse Up*) σε οποιοδήποτε σημείο της οθόνης το πρόγραμμα αποθηκεύει προσωρινά σε μια καινούργια μεταβλητή τύπου *Point* τις καινούργιες συντεταγμένες του δείκτη του ποντικιού στην οθόνη και αφαιρώντας τις αντίστοιχες τιμές (*X,Y*) των συντεταγμένων της αρχικής θέσης έχουμε τις συντεταγμένες της καινούργιας θέσης της κλάσης μας στην οθόνη.

WidgetContextMenuStrip Items Functions

Οι μέθοδοι αυτής της κατηγορίας αφορούν τις λειτουργίες του μενού επιλογών που εμφανίζεται με το πάτημα του δεξιού πλήκτρου του ποντικιού τόσο μέσα στη κλάση όσο και μέσα στο εικονίδιο της εφαρμογής που εμφανίζεται στο System Tray.

Drag and Drop Functions

Οι μέθοδοι αυτής της κατηγορίας ασχολούνται με τη λειτουργία της κλάσης WidgetForm κατά τη διάρκεια του Drag And Drop ενός αντικείμενου (αρχείου ή φακέλου) στην κλάση. Όταν γίνει Drop ένα αντικείμενο στη κλάση, ελέγχεται αρχικά αν είναι αρχείο ή φάκελος. Αν είναι αρχείο τότε η κλάση, χρησιμοποιώντας τη μέθοδο Copy της κλάσης File του System.IO, το αντιγράφει στο φάκελο του χρήστη που βρίσκεται στην επιφάνεια εργασίας. Αν το αντικείμενο είναι φάκελος τότε χρησιμοποιείται η μέθοδος DropDirectory, της οποίας η λειτουργία θα αναλυθεί παρακάτω, για την αντιγραφή όλων των περιεχομένων στον φάκελο του χρήστη.

4.3.4.4 Οι Μέθοδοι της κλάσης LoginForm

Οι μέθοδοι της κλάσης, εκτός από τη μέθοδο DropDirectory η οποία ασχολείται με την αντιγραφή στο φάκελο του χρήστη των περιεχομένων ενός φακέλου που γίνεται drop στη κλάση, ασχολούνται αποκλειστικά με την ανίχνευση των τροποποιημένων αρχείων ή φακέλων στον φάκελο του χρήστη στην επιφάνεια εργασίας και την ορθή ενημέρωση του FTP Server για τις τροποποιήσεις αυτές.

```
Private Sub DropDirectory(ByVal SourceDirectory As String, ByVal DestinationDirectory As String)
```

Η μέθοδος DropDirectory δέχεται σαν ορίσματα (τύπου String) το μονοπάτι του φακέλου που επιθυμούμε να αντιγράψουμε (SourceDirectory) και το μονοπάτι του φακέλου στον οποίο θα γίνει η αντιγραφή (DestinationDirectory). Αρχικά δημιουργεί μια μεταβλητή τύπου DirectoryInfo η οποία περιέχει όλες τις πληροφορίες του προς αντιγραφή φακέλου (SourceDirectory) και ελέγχει εάν υπάρχει ο φάκελος που θα γίνει η αντιγραφή (DestinationDirectory). Χρησιμοποιώντας τις μεθόδους DirectoryInfo.GetFiles, η οποία επιστρέφει μία λίστα τύπου FileInfo με όλα τα αρχεία που υπάρχουν μέσα στο φάκελο που καλεί τη μέθοδο, και DirectoryInfo.GetDirectories, η οποία επιστρέφει μία λίστα τύπου DirectoryInfo με όλους τους υποφακέλους που υπάρχουν μέσα στο φάκελο που καλεί τη μέθοδο, η μέθοδος αντιγράφει όλα τα αρχεία και δημιουργεί όλους τους φακέλους που υπάρχουν στο SourceDirectory. Η μέθοδος καλεί τον εαυτό της για κάθε στοιχείο της λίστας που επιστρέφει η DirectoryInfo.GetDirectories επαναλαμβάνοντας τη διαδικασία για κάθε υποφάκελο του SourceDirectory.

```
Private Sub DropDirectory(ByVal SourceDirectory As String, ByVal DestinationDirectory As String)
```

```
Dim CurrentDirectory As DirectoryInfo = New DirectoryInfo(SourceDirectory)
```

```

If Not Directory.Exists(DestinationDirectory) Then
    Directory.CreateDirectory(DestinationDirectory)
End If

For Each CurrentFileInfo In CurrentDirectory.GetFiles()
    Try
        CurrentFileInfo.CopyTo(Path.Combine(DestinationDirectory, CurrentFileInfo.Name))
    Catch ex As Exception
        File.Create(Path.Combine(DestinationDirectory, CurrentFileInfo.Name))
    End Try
Next

For Each CurrentDirectoryInfo In CurrentDirectory.GetDirectories()
    Dim SubDirectory As String = Path.Combine(DestinationDirectory, _
                                                CurrentDirectoryInfo.Name)

    DropDirectory(CurrentDirectoryInfo.FullName, SubDirectory)
Next
End Sub

```

Private Sub DeleteFromFTPServer()

Η μέθοδος DeleteFromFTPServer είναι υπεύθυνη για τον εντοπισμό όλων των διεγραμμένων αρχείων ή φακέλων στον φάκελο του χρήστη και της αποστολή των κατάλληλων εντολών στον FTP Server για την υλοποίηση των αλλαγών αυτών και στον FTP φάκελο. Για τον εντοπισμό των διεγραμμένων αρχείων ή φακέλων χρησιμοποιείται η λίστα περιεχομένων που δημιουργήθηκε κατά την εκκίνηση της κλάσης (IndexerOnLoad) και η λίστα περιεχομένων που δημιουργήθηκε απο την λειτουργία που καλεί τη μέθοδο (IndexerOnClose). Η μέθοδος αρχικά ελέγχει για τυχόν διεγραμμένα αρχεία κάνοντας σύγκριση των αντικειμένων των δυο λιστών και σε περίπτωση που κάποιο αντικείμενο της λίστας του IndexerOnLoad δεν υπάρχει στη λίστα του IndexerOnClose τότε, χρησιμοποιώντας τη μέθοδο DeleteFile της κλάσης FtpOperations, η μέθοδος στέλνει την εντολή διαγραφής του αρχείου απο τον φάκελο του FTP Server.

```

For Each IndexerOnLoadItem In IndexerOnLoad.GetFiles

For Each IndexerOnCloseItem In IndexerOnClose.GetFiles

If IndexerOnCloseItem.FullName = IndexerOnLoadItem.FullName Then

    GoTo TryNextFile

End If

Next

FTPConnection.DeleteFile(IndexerOnLoadItem.FullName)

TryNextFile: Next

```

Όταν ολοκληρωθεί ο έλεγχος για τυχόν διεγραμμένα αρχεία τότε ξεκινάει ο έλεγχος για τυχόν διεγραμμένους φακέλους. Αρχικά δημιουργούνται δυο λίστες τύπου DirectoryInfo στις οποίες αποθηκεύονται οι λίστες που επιστρέφει η μέθοδος GetDirectories (που βρίσκεται στην κλάση Indexer) των IndexerOnLoad και IndexerOnClose. Οι λίστες είναι ταξινομημένες αλφαβητικά, που σημαίνει ότι οι υποφάκελοι σαν αντικείμενα της λίστας θα βρίσκονται πάντα μετά το φάκελο στον οποίο βρίσκονται. Η μέθοδος αντιστρέφοντας τις λίστες, χρησιμοποιώντας τη μέθοδο Reverse, επιτυγχάνει τον εντοπισμό και διαγραφή αρχικά όλων των υποφακέλων ενός διεγραμμένου φακέλου, αποφεύγοντας έτσι το σφάλμα που προκύπτει από την προσπάθεια διαγραφής από τον FTP Server ενός φακέλου που δεν είναι άδειος. Η μέθοδος ελέγχει για τυχόν διεγραμμένους φακέλους κάνοντας σύγκριση των αντικειμένων των δυο λιστών και σε περίπτωση που κάποιο αντικείμενο της μιας λίστας δεν υπάρχει στην άλλη λίστα τότε, χρησιμοποιώντας τη μέθοδο RemoveDirectory της κλάσης FtpOperations, η μέθοδος στέλνει την εντολή διαγραφής του φακέλου από τον φάκελο του FTP Server.

```

Dim IndexerOnLoadDirectories As List(Of DirectoryInfo) = IndexerOnLoad.GetDirectories

Dim IndexerOnCloseDirectories As List(Of DirectoryInfo) = IndexerOnClose.GetDirectories

IndexerOnLoadDirectories.Reverse()

IndexerOnCloseDirectories.Reverse()

For Each IndexerOnLoadItem In IndexerOnLoadDirectories

    For Each IndexerOnCloseItem In IndexerOnCloseDirectories

        If IndexerOnCloseItem.FullName = IndexerOnLoadItem.FullName Then

            GoTo TryNextDirectory

        End If

    Next

FTPConnection.RemoveDirectory(IndexerOnLoadItem.FullName)

TryNextDirectory: Next

```

```

Private Sub CreateToFTPServer()

```

Η μέθοδος CreateToFTPServer είναι υπεύθυνη για τον εντοπισμό όλων των καινούργιων αρχείων ή φακέλων στον φάκελο του χρήστη και της αποστολή των κατάλληλων εντολών στον FTP Server για την υλοποίηση των αλλαγών αυτών και στον FTP φάκελο. Για τον εντοπισμό των καινούργιων αρχείων ή φακέλων χρησιμοποιείται η λίστα περιεχομένων που δημιουργήθηκε κατά την εκκίνηση της κλάσης (IndexerOnLoad) και η λίστα περιεχομένων που δημιουργήθηκε από την λειτουργία που καλεί τη μέθοδο (IndexerOnClose). Η μέθοδος αρχικά ελέγχει για τυχόν καινούργιους φακέλους κάνοντας σύγκριση των αντικειμένων των δυο λιστών και σε περίπτωση που κάποιο αντικείμενο της λίστας του IndexerOnClose δεν υπάρχει στη λίστα του IndexerOnLoad τότε,

χρησιμοποιώντας τη μέθοδο MakeDirectory της κλάσης FtpOperations, η μέθοδος στέλνει την εντολή δημιουργίας του φακέλου στον φάκελο του FTP Server.

Σε περίπτωση επανεκκίνησης της εφαρμογής μετά απο μη κανονικό τερματισμό το πρόγραμμα διατηρεί στο φάκελο του χρήστη όλα τα αρχεία ή φακέλους που προστέθηκαν μετά την τελευταία κανονική εκκίνηση της εφαρμογής. Το πρόβλημα που δημιουργείται σε αυτή τη περίπτωση είναι ότι όλα αυτά τα αρχεία και οι φάκελοι δεν υπάρχουν στον FTP Server αλλά υπάρχουν στη λίστα περιεχομένων που δημιουργήθηκε κατά την εκκίνηση της κλάσης (IndexerOnLoad) με αποτέλεσμα μην τα αναγνωρίζει η μέθοδος σαν καινούργια, εφόσον υπάρχουν και στις δυο λίστες, και επομένως να μη γίνεται σωστή ενημέρωση του FTP Server. Για το λόγο αυτό κατά την εκκίνηση της μεθόδου, χρησιμοποιώντας τη μέθοδο IndexLogFullPathList του module PublicVariables, δημιουργείται μια λίστα τύπου String (IndexLogList) με τα περιεχόμενα του αρχείου IndexLog.nfo (περιεχόμενα του φακέλου του FTP Server) που υπάρχουν στον φάκελο του χρήστη. Έτσι αν ένα αντικείμενο υπάρχει και στις δυο λίστες (IndexerOnLoad και IndexerOnClose) και δεν υπάρχει στη λίστα IndexLogList τότε σημαίνει ότι πρέπει να δημιουργηθεί (αν πρόκειται για φάκελο) ή να αποσταλλεί (αν πρόκειται για αρχείο) στον FTP Server.

```
Dim IndexLogList As List(Of String) = _
    PublicVariables.IndexLogFullPathList(FTPConnection.GetUsername)

For Each IndexerOnCloseItem In IndexerOnClose.GetDirectories

    For Each IndexerOnLoadItem In IndexerOnLoad.GetDirectories

        If IndexerOnLoadItem.FullName = IndexerOnCloseItem.FullName Then

            If IndexLogList.Contains(IndexerOnLoadItem.FullName & "\") Then

                GoTo TryNextDirectory

            End If

        End If

    End If

Next
```



```
FTPConnection.MakeDirectory(IndexerOnCloseItem.FullName)
```

```
TryNextDirectory: Next
```

Μετά την ολοκλήρωση του ελέγχου για τυχόν καινούργιους φακέλους, ξεκινάει ο έλεγχος για τυχόν καινούργια αρχεία. Η μέθοδος ελέγχει για τυχόν καινούργια αρχεία κάνοντας σύγκριση των αντικειμένων των δυο λιστών (IndexerOnLoad και IndexerOnClose) και σε περίπτωση που κάποιο αντικείμενο της λίστας του IndexerOnClose δεν υπάρχει στη λίστα του IndexerOnLoad τότε, χρησιμοποιώντας τη μέθοδο UploadFile της κλάσης FtpOperations, η μέθοδος στέλνει την εντολή ανεβάσματος του αρχείου στον φάκελο του FTP Server. Σε περίπτωση που κάποιο αντικείμενο της λίστας του IndexerOnClose υπάρχει στη λίστα του IndexerOnLoad η μέθοδος ελέγχει την ημερομηνία και ώρα τελευταίας τροποποίησης και αν δεν είναι ίδια σημαίνει ότι το αρχείο τροποποιήθηκε κατά τη διάρκεια της λειτουργίας της εφαρμογής και πρέπει να αποσταλλεί στον FTP Server. Επίσης αν κάποιο αντικείμενο υπάρχει και στις δυο λίστες και δεν υπάρχει στη λίστα IndexLogList τότε σημαίνει ότι το αρχείο δημιουργήθηκε ή προστέθηκε στο φάκελο μετά την τελευταία κανονική εκκίνηση της εφαρμογής οπότε πρέπει να αποσταλλεί στον FTP Server.

```
For Each IndexerOnCloseItem In IndexerOnClose.GetFiles
```

```
For Each IndexerOnLoadItem In IndexerOnLoad.GetFiles
```

```
If IndexerOnLoadItem.FullName = IndexerOnCloseItem.FullName Then
```

```
If Not IndexerOnLoadItem.LastWriteTime = _
```

```
File.GetLastWriteTime(IndexerOnCloseItem.FullName) Then
```

```
FTPConnection.UploadFile(IndexerOnCloseItem.FullName)
```

```
End If
```

```
If IndexLogList.Contains(IndexerOnLoadItem.FullName) Then
```

```
GoTo TryNextFile
```

```
End If
```

End If

Next

```
FTPConnection.UploadFile(IndexerOnCloseItem.FullName)
```

TryNextFile: [Next](#)

Κεφάλαιο 5

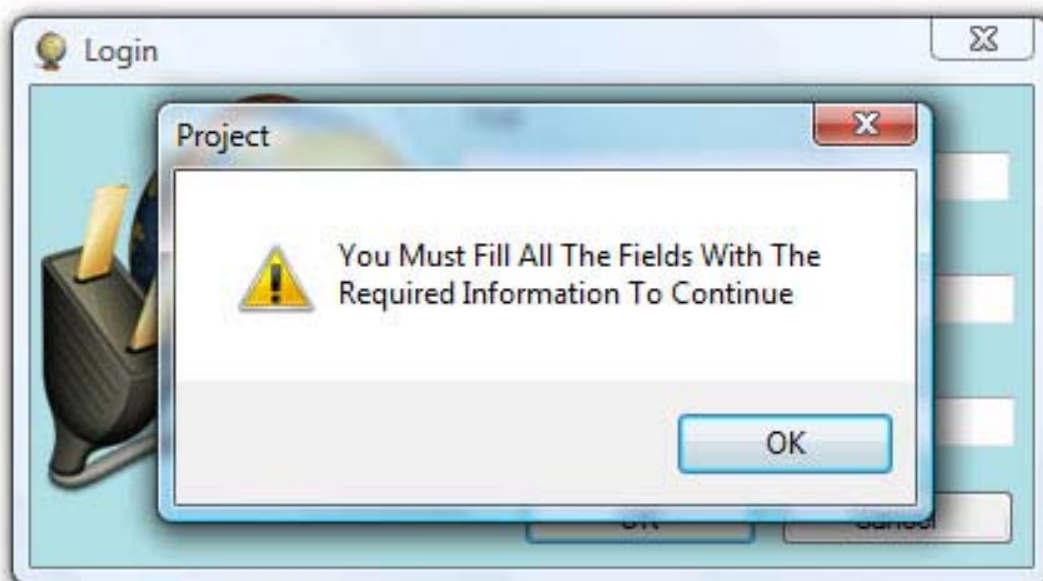
Εγχειρίδιο Χρήσης

Κατα την εκκίνηση του προγράμματος εμφανίζεται το παράθυρο Login το οποίο προτρέπει τον χρήστη να εισάγει τα απαραίτητα στοιχεία για τη σύνδεση με τον απομακρυσμένο διακομιστή FTP. Στο πεδίο του εισάγουμε τη διεύθυνση του απομακρυσμένου διακομιστή FTP ,στο πεδίο εισάγουμε το όνομα χρήστη και στο πεδίο εισάγουμε τον κωδικό πρόσβασης.



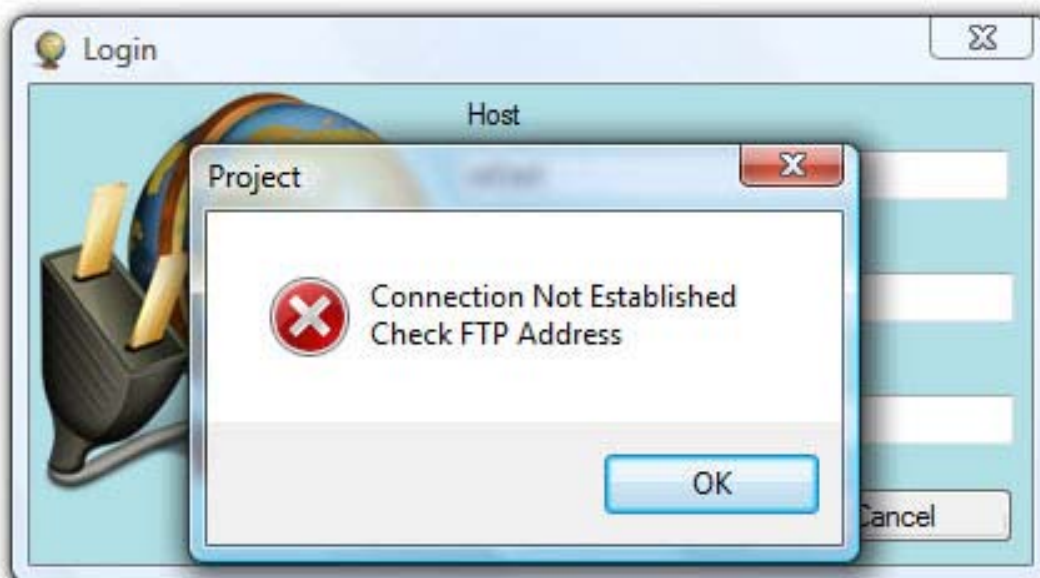
Εικόνα 5.1 Το παράθυρο Login

Σε περίπτωση που υπάρχει άδειο πεδίο το πρόγραμμα τον χρήστη να συμπληρώσει τα στοιχεία που λείπουν και επιστρέφει στην αρχική κατάσταση.



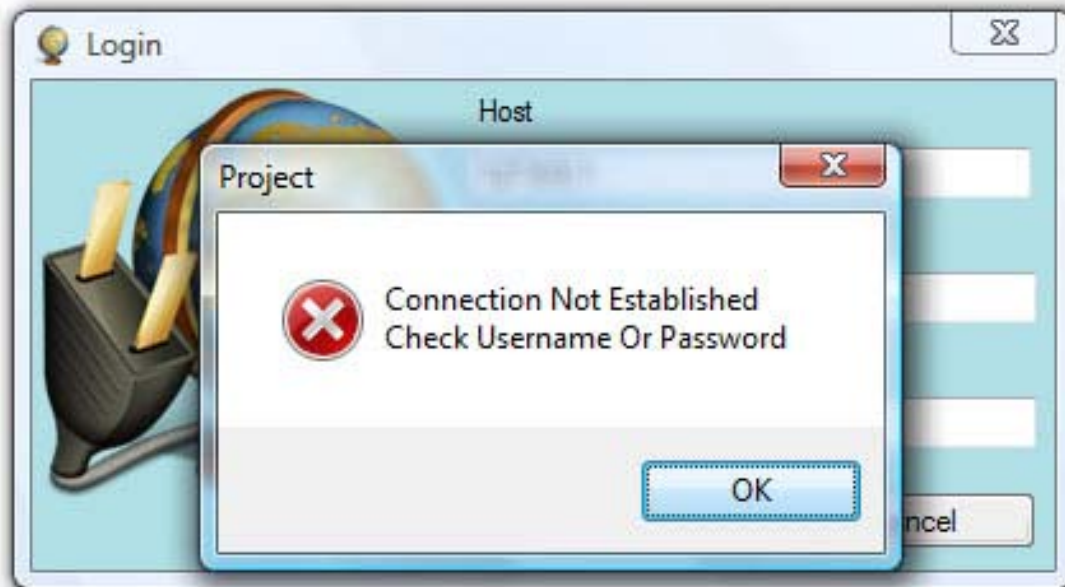
Εικόνα 5.2 Παράθυρο προειδοποίησης άδειων πεδίων

Σε περίπτωση που η διεύθυνση του απομακρυσμένου διακομιστή δεν είναι σωστή το πρόγραμμα ενημερώνει τον χρήστη και επιστρέφει στην αρχική κατάσταση.



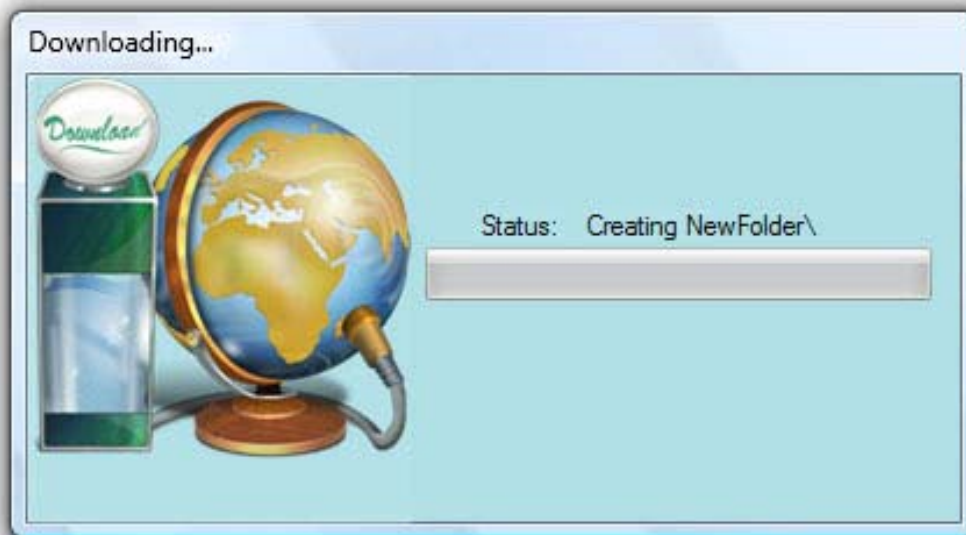
Εικόνα 5.3 Παράθυρο προειδοποίησης λανθασμένης διεύθυνσης FTP

Σε περίπτωση που το όνομα χρήστη ή ο κωδικός πρόσβασης δεν είναι σωστά το πρόγραμμα ενημερώνει τον χρήστη και επιστρέφει στην αρχική κατάσταση.



Εικόνα 5.4 Παράθυρο προειδοποίησης λανθασμένου ονόματος χρήστη ή κωδικου προσβασης

Αν γίνει επιτυχημένη σύνδεση με τον απομακρυσμένο διακομιστή FTP τότε το πρόγραμμα εξαφανίζει το παράθυρο Login και ξεκινάει η διαδικασία συγχρονισμού του φακέλου του απομακρυσμένου διακομιστή με την περιοχή του χρήστη στον τοπικό υπολογιστή. Το πρόγραμμα σ' αυτή τη φάση εμφανίζει το παρακάτω παράθυρο το οποίο μας δείχνει τους φακέλους και τα αρχεία που δημιουργούνται στον τοπικό υπολογιστή.



Εικόνα 4.3 Δημιουργία φακέλου



Εικόνα 4.4 Κατέβασμα αρχείου

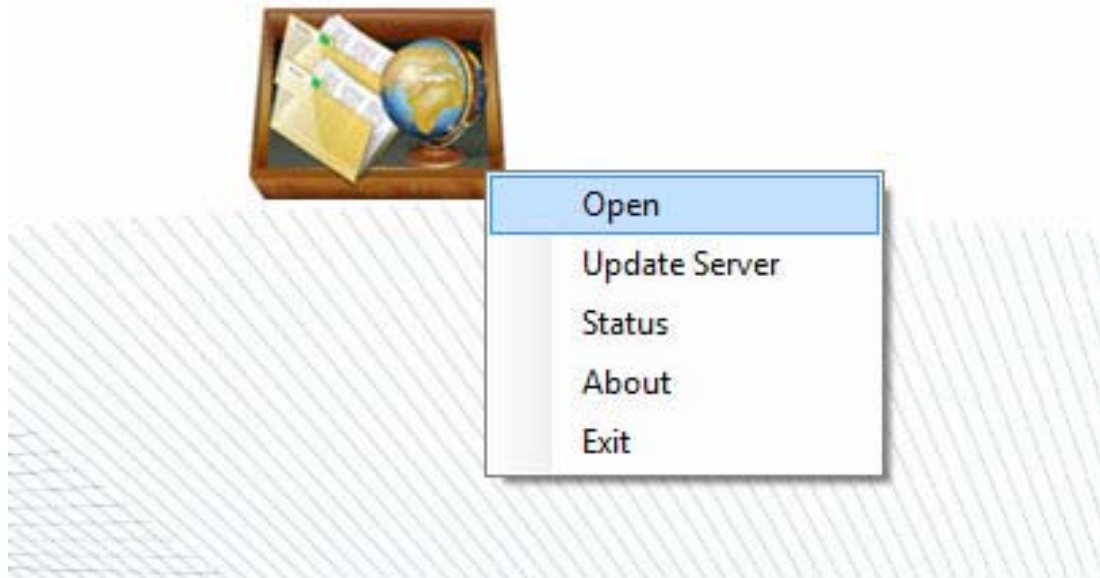
Με την ολοκλήρωση της διαδικασίας το πρόγραμμα εξαφανίζει το παραπάνω παράθυρο και εμφανίζει το παρακάτω εικονίδιο.



Εικόνα 5.4 Το εικονίδιο του προγράμματος

Κάνοντας διπλό κλικ με το αριστερό πλήκτρο του ποντικιού μας πάνω στο εικονίδιο το πρόγραμμα ανοίγει την περιοχή του χρήστη στον τοπικό υπολογιστή, όπου βρίσκονται όλα τα αρχεία που υπάρχουν στον φάκελο του απομακρυσμένου διακομιστή FTP.

Ένας άλλος τρόπος για να ανοίξει η περιοχή του χρήστη στον τοπικό υπολογιστή είναι χρησιμοποιώντας το μενού επιλογών που εμφανίζεται κάνοντας δεξί κλικ με το ποντίκι μας πάνω στο εικονίδιο και επιλέγοντας Open.



Εικόνα 5.5 Το μενού επιλογών του προγράμματος

Η επιλογή Update Server του μενού επιλογών, πραγματοποιεί τον συγχρονισμό του τοπικού υπολογιστή και του απομακρυσμένου διακομιστή FTP, κατά τον οποίο το πρόγραμμα ενημερώνει τον φάκελο του απομακρυσμένου διακομιστή FTP για όλες τις αλλαγές που πραγματοποιήθηκαν στην περιοχή του χρήστη στον τοπικό υπολογιστή.

Η επιλογή Status του μενού επιλογών εμφανίζει ένα παράθυρο με πληροφορίες σχετικά με τη σύνδεση που πραγματοποιήθηκε.

Η επιλογή About του μενού επιλογών εμφανίζει ένα παράθυρο με πληροφορίες σχετικά με το πρόγραμμα.

Τέλος, η επιλογή Exit τερματίζει το πρόγραμμα. Κατά τη διάρκεια του τερματισμού το πρόγραμμα ενημερώνει τον φάκελο του απομακρυσμένου διακομιστή FTP για όλες τις αλλαγές που πραγματοποιήθηκαν στην περιοχή του χρήστη στον τοπικό υπολογιστή.

Βιβλιογραφία

- Δίκτυα Υπολογιστών – Andrew S. Tanenbaum (*Εκδόσεις Παπασωτηρίου*).
- Διαδραστικές Εφαρμογές Πολυμέσων / Τεχνολογία, σχεδιασμός και διαδικασίες υλοποίησης – Φράνκα Παντάνο-Ρόκου (*Εκδόσεις Κριτική*).
- Μικρός Θησαυρός Πληροφορικής / Βασικό ερμηνευτικό λεξικό – Φοίβος Αρβανίτης (*Εκδόσεις Αλκυών*).
- Visual Basic .NET Developer's Handbook - Evangelos Petroutsos, Mark Ridgeway (Εκδόσεις Sybex).
- Mastering Microsoft Visual Basic 2008 - Evangelos Petroutsos (Εκδόσεις Sybex).
- <http://msdn.microsoft.com>
- <http://msdn.microsoft.com/el-gr/default.aspx>
- <http://www.dotnetzone.gr>
- <http://www.codeproject.com>