



**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

ΦΟΙΤΗΤΗΣ:

ΛΥΓΚΟΥΝΑΚΗΣ ΕΥΑΓΓΕΛΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ : ΚΑΤΑΣΚΕΥΗ RICH INTERNET APPLICATION (RIA) ΒΑΣΙΣΜΕΝΗ ΣΕ
ΧΑΡΤΕΣ ΣΕ ADOBE FLEX ΚΑΙ AIR.



ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ : Κ. ΜΑΛΑΜΟΣ ΑΘΑΝΑΣΙΟΣ

Ηράκλειο 2010

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	4
ΚΕΦΑΛΑΙΟ 1 Εισαγωγή	6
1.1 Κατασκευή Rich Internet Application (RIA) βασισμένη σε χάρτες, με την βοήθεια του λογισμικού Adobe Flex και Adobe Air	6
1.2 Αντικείμενο Πτυχιακής	6
1.2.1 Συνεισφορά	7
1.2.2 Οργάνωση κειμένου	7
ΚΕΦΑΛΑΙΟ 2 Σχετικές Εργασίες	8
2.1 Παραδείγματα από mashups	8
2.1.1 Flash Earth	8
2.1.2 Flappr	9
2.1.3 Διαδίδοντας μια διαδικτυακή εφαρμογή	11
2.1.4 Feeds	11
ΚΕΦΑΛΑΙΟ 3 Θεωρητικό Υπόβαθρο	15
3.1 APIs	15
3.2 Τα πλεονεκτήματα μιας open εφαρμογής	16
3.2.1 Η καινοτομία ενθαρρύνεται	16
3.2.2 Επέκταση της λειτουργικότητας	16
3.2.3 Αύξηση της κυκλοφορίας	16
3.2.4 Κοινότητες	17
3.2.5 Ανοικτά δεδομένα	17
3.2.6 Εσωτερικά mashups	17
3.2.7 Ευκαιρίες κέρδους χρημάτων	18
3.3 Εισαγωγή στο Flex	18
3.3.1 Χρησιμοποιώντας Design Mode	19
3.3.2 Χρησιμοποιώντας Source Mode	31
3.4 Κατασκευή μιας Flex εφαρμογής	35
3.4.1 Δημιουργία νέου Flex project	36
3.4.2 Ορίζοντας την εφαρμογή σας	37
3.4.3 Ολοκλήρωση των δεδομένων σας	38
3.5 Περνώντας στο Desktop	42
3.5.1 Γιατί να αναπτύξω εφαρμογή desktop;	45
3.6 Δημιουργία AIR εφαρμογής	46
3.6.1 Ξεκινώντας νέο project	47
3.6.2 Καθορισμός και προσαρμογή εφαρμογής	52
3.6.2.1 Τροποποιώντας το παράθυρο εφαρμογής	53
3.6.2.2 Εργασία με διαφορετικούς τύπους περιεχομένου	54
3.6.2.3 Προσθέτοντας εικόνες και οι φάκελους	55
3.7 Διανομή μιας AIR εφαρμογής	56
3.7.1 Εγκατάσταση AIR εφαρμογής από την οπτική γωνία του χρήστη	56
3.7.2 Δημιουργία AIR αρχείου	59
ΚΕΦΑΛΑΙΟ 4 Ανάλυση Απαιτήσεων Συστήματος	64
4.1 Εισαγωγή	64
4.1.1 Τι είναι Google Maps API για Flash;	64
4.1.2 Εγγραφή για Google Maps API Key	65
4.1.3 Κατέβασμα του Google Maps API για Flash SDK	65
4.1.4 Κατασκευή Απλού χάρτη	65

4.1.5 Προσθήκη του API Key.	65
4.2 Events.	65
4.2.1 Event Listeners.	65
4.3 Controls.	66
4.3.1 Προσθήκη Controls στον χάρτη.	66
4.3.2 Ρύθμιση Controls.	67
4.3.3 Τοποθέτηση Controls στο χάρτη.	68
4.4 Overlays	68
4.4.1 Markers.	69
4.4.2 Polygons.	70
4.4.3 Ground Overlays	71
4.4.4 InfoWindows	72
4.5 Services.	73
4.5.1 Geocoding	73
ΚΕΦΑΛΑΙΟ 5 Σχεδίαση Συστήματος	75
5.1 Αρχιτεκτονική.	75
5.2 Περιγραφή Κλάσεων και συναρτήσεων.	80
ΚΕΦΑΛΑΙΟ 6 Έλεγχος	85
6.1 Αναλυτική παρουσίαση ελέγχου.	85
ΚΕΦΑΛΑΙΟ 7 Υλοποίηση	90
7.1 Πλατφόρμες και προγραμματιστικά εργαλεία.	90
7.1.1 Adobe Flash Platform.	90
7.1.2 Adobe Flex Builder version 3.	90
7.1.3 Mxml.	91
7.2 Διαδικασία εγκατάστασης σε υπολογιστή.	91
ΚΕΦΑΛΑΙΟ 8 Βιβλιογραφία	93

ΠΕΡΙΛΗΨΗ

Μπορούμε να πούμε ότι ένας από τους ταχύτερα εξελισσόμενους τομείς της τεχνολογίας, όσον αφορά την καινοτομία είναι το Web. Σε διάστημα δέκα ετών, το web έχει περάσει από το να περιέχει μόνο κείμενο, σε μια πλατφόρμα πάνω στην οποία οι προγραμματιστές μπορούν να δημιουργήσουν συναρπαστικές και άκρως διαδραστικές εφαρμογές. Οι επιπτώσεις στην επιχειρηματική δραστηριότητα, ψυχαγωγία και καθημερινές εργασίες είναι τεράστιες. Και ακριβώς τη στιγμή που πολλοί άνθρωποι σκέφτονται να είχαμε φτάσει στο ανώτατο σημείο, ήρθε η έννοια mashups, εφαρμογές που συνδυάζουν πληροφορίες από περισσότερες από μία πηγή σε μία και μόνο εφαρμογή. Επιτρέπουν στους χρήστες να βλέπουν τα δεδομένα με νέους και πιο χρήσιμους τρόπους, και τους επιτρέπουν να χρησιμοποιήσουν την φαντασία τους. Το Adobe Flex και το Adobe AIR παρέχουν μια ισχυρή βάση που επιτρέπει στους προγραμματιστές γρήγορα και εύκολα να δημιουργήσουν εφαρμογές mashup με toolsets και γνώσεις που ήδη χρησιμοποιούνται. Η πτυχιακή εργασία στοχεύει στη σχεδίαση και εφαρμογή μιας πλούσιας διαδικτυακής εφαρμογής, που θα κατασκευαστεί σε Adobe Flex / Air.

ABSTRACT

We can say that one of the most rapidly evolving fields of technology, as innovation is the Web .In ten years, the web has gone from only contain text, a platform on which developers can create exciting and highly interactive applications. The impact on business, entertainment and daily tasks are enormous. And just when many people think we had reached the peak, came the concept of mashups, applications that combine information from more than one source into a single application. Allow users to view data in new and more useful ways, and allow them to use their imagination. The Adobe Flex and Adobe AIR provide a solid platform that enables developers to quickly and easily create applications mashup with toolsets and knowledge already in use. This thesis aims to design and implement a rich web application, which will be built in Adobe Flex / Air.

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Κατασκευή Rich Internet Application (RIA) βασισμένη σε χάρτες, με την βοήθεια του λογισμικού Adobe Flex και Adobe Air.

Η προτεινόμενη εργασία στοχεύει στη σχεδίαση και εφαρμογή μιας πλούσιας διαδικτυακής εφαρμογής, που θα κατασκευαστεί σε Adobe Flex / Air. Η εφαρμογή αυτή θα είναι βασισμένη στη γλώσσα προγραμματισμού Mxml και θα καλύπτει το κομμάτι της χαρτογράφησης με τη χρησιμοποίηση του καινούριου Maps API της Google για Flash. Θα καλυφθούν τα κομμάτια: Events, Map Control, Overlays όπως είναι τα Markers, τα Poly-lines, τα Polygons και Services όπως Geo-coding. Η τελική εφαρμογή θα είναι σε τεχνολογία Flash η οποία είναι συμβατή με πάνω από το 95% των σύγχρονων υπολογιστικών συστημάτων.

1.2 Αντικείμενο Πτυχιακής

Η συγκεκριμένη εφαρμογή που θα δημιουργηθεί θα έχει ως βάση το Google Maps Api και θα κατασκευαστεί με την βοήθεια του Adobe Flex. Κατά την εκκίνηση του προγράμματος θα ενημερώνεται η εφαρμογή από ένα αρχείο γραμμένο σε γλώσσα προγραμματισμού Xml σε ποια σημεία του χάρτη της Κρήτης υπάρχουν καταστήματα τραπεζών και θα προσθέτονται αυτόματα markers σε αυτά τα σημεία. Ο χάρτης θα διαθέτει τα απαραίτητα controls έτσι ώστε να μπορεί ο χρήστης να πλοηγείται με ευκολία στον χάρτη. Στο κάτω μέρος της εφαρμογής θα υπάρχει μια ορατή βάση δεδομένων των τραπεζών από την οποία ο χρήστης θα μπορεί να κάνει εύκολα ταξινόμηση σύμφωνα με διάφορα δεδομένα όπως ταξινόμηση ανά περιοχή, ανά ταχυδρομικό κώδικα κ.α. Ακόμα ο χρήστης θα έχει την δυνατότητα να βλέπει διαδραστικά πάνω στον χάρτη μέσω ενός menu επιλογής με ποιες εταιρίες συνεργάζεται η κάθε τράπεζα σε εσωτερικό και εξωτερικό (visa, cirus, maestro κοκ.).

1.2.1 Συνεισφορά

1. Έγινε μελέτη των συστημάτων Adobe Flex και Adobe Air καθώς και του Maps API της Google
2. Υλοποιήσαμε αλγορίθμους για τη λειτουργία της εφαρμογής, χρησιμοποιήσαμε interfaces, datagrid και ανάγνωση δεδομένων από εξωτερικό xml αρχείο.
3. Αξιολογήσαμε την επίδοση της εφαρμογής και βρήκαμε ότι είναι ικανοποιητική για την πλειοψηφία των σύγχρονων πληροφοριακών συστημάτων.
4. Ενσωματώσαμε τους αλγορίθμους σε πρότυπο σύστημα γραμμένο σε γλώσσα προγραμματισμού Xml.
5. Έγινε έλεγχος για την σωστή λειτουργία της εφαρμογής.

1.3 Οργάνωση κειμένου

Το Κεφάλαιο 1 είναι η εισαγωγή της πτυχιακής εργασίας. Γίνεται γενική αλλά και συγκεκριμένη περιγραφή του χώρου εφαρμογής της πτυχιακής εργασίας. Εργασίες σχετικές με το αντικείμενο της πτυχιακής παρουσιάζονται στο Κεφάλαιο 2 . Το Κεφάλαιο 3 αναφέρεται στο θεωρητικό υπόβαθρο της εργασίας όπως είναι τα πλεονεκτήματα μιας open εφαρμογής, τη διαδικασία κατασκευής μιας εφαρμογής σε Adobe Flex και Air, ορισμοί καθώς και πληροφορίες που αφορούν την διανομή εφαρμογών. Στο Κεφάλαιο 4 γίνεται η ανάλυση των απαιτήσεων του συστήματος, αναλύοντας κάποιες βασικές έννοιες γύρω από τον προγραμματισμό του Maps Api της Google σε γλώσσα Xml. Το Κεφάλαιο 5 αναφέρεται στη σχεδίαση του συστήματος και συγκεκριμένα στην αρχιτεκτονική του παρουσιάζοντας ορισμένα κομμάτια στα οποία έχει χτιστεί πάνω ο κώδικας μας και περιγράφοντας τις κλάσεις, τις μεθόδους και τα interface που χρησιμοποιήθηκαν για την κατασκευή της εφαρμογής. Στο Κεφάλαιο 6 παρουσιάζουμε αναλυτικά τον έλεγχο του συστήματος Το κεφάλαιο αυτό, λειτουργεί και ως εγχειρίδιο χρήσης του προγράμματος. Στο Κεφάλαιο 7 είναι ο επίλογος της πτυχιακής. Συνοψίζουμε τα αποτελέσματα της διπλωματικής και περιγράφουμε τα συμπεράσματα που προέκυψαν, αρνητικά και θετικά και δίνουμε ιδέες για επέκταση της εφαρμογής. Τέλος στο Κεφάλαιο 8 αναφέρουμε την βιβλιογραφία.

ΚΕΦΑΛΑΙΟ 2 Σχετικές Εργασίες

2.1 Παραδείγματα από mashups

Υπάρχουν πολλές διαφορετικές ερμηνείες για το τι είναι mashups και τι σημαίνουν για το Web. Σε γενικές γραμμές, ένα mashup είναι μια εφαρμογή που λαμβάνει δεδομένα από διαφορετικές πηγές και τα συνδυάζει προς όφελος του χρήστη. Έτσι αυτό μπορεί να προσθέσει αξία σε προϋπάρχοντα δεδομένα. Με πολλούς τρόπους, τα mashups προσφέρουν μια καινοτόμο πλατφόρμα, όπου οι προγραμματιστές μπορούν να αποδείξουν το ταλέντο και η δημιουργικότητα τους χωρίς να χρειάζεται να επανεφεύρουν τον τροχό, και να ψάχνουν πρόσβαση σε δεδομένα που μπορεί και να μην είναι στη διάθεσή τους.

Αν και αρχικά τα mashups ήταν ένας όρος που χρησιμοποιούταν για να περιγράψει web εφαρμογές που χτίστηκαν γύρω από μια ή περισσότερες πηγές δεδομένων, εμείς τα βλέπουμε να μετακινούνται από το διαδίκτυο στην επιφάνεια εργασίας ως πλατφόρμες widget.

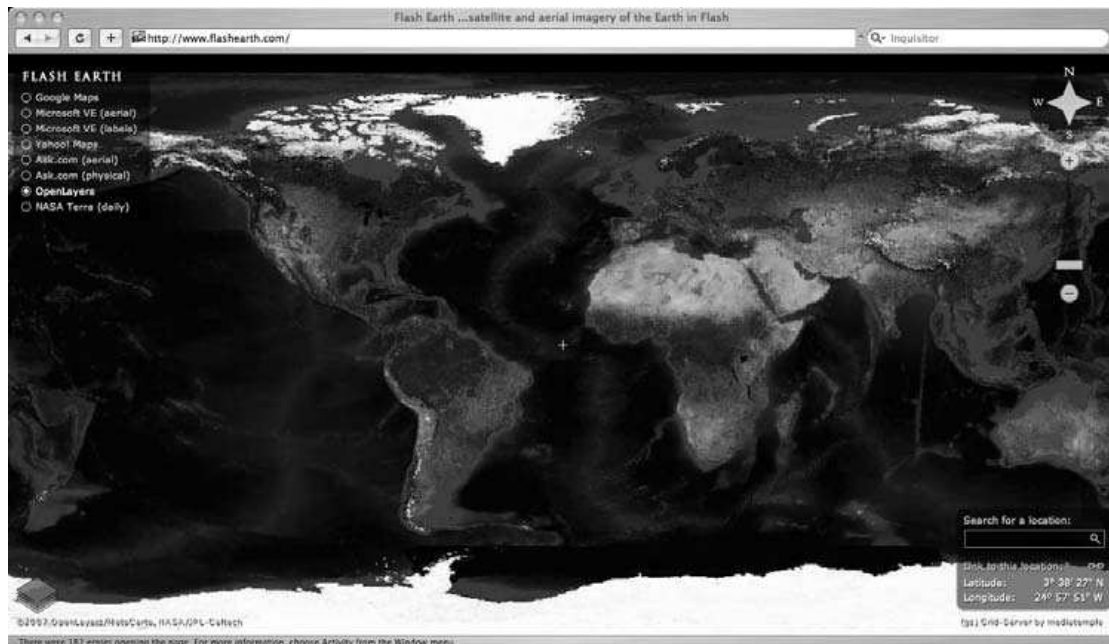
Τα δεδομένα μπορούν λαμβάνονται από διάφορες πηγές, συμπεριλαμβανομένων των εξής:

- Public APIs
- Web services
- Data feeds (such as RSS)

Τα Mashups έχουν αγκαλιαστεί από πολλές εταιρείες του Διαδικτύου, συμπεριλαμβανομένων των Google, Yahoo και Amazon, οι οποίες αναζητούσαν διάφορες μεθόδους για την ανάπτυξη και λειτουργικότητα των εφαρμογών στις ιστοσελίδες τους.

2.1.1 Flash Earth (<http://www.flashearth.com/>)

Το Flash Earth είναι ένα ενδιαφέρον παράδειγμα, αντί να συνδυάζει διάφορους τύπους δεδομένων, επιτρέπει στο χρήστη να επιλέγει μεταξύ διαφορετικών πηγών των ίδιων στοιχείων. Στην περίπτωση αυτή, η αίτηση μπορεί να ανακτήσει δεδομένα από εκτεταμένη συλλογή χαρτών και αεροφωτογραφιών σε Google, Yahoo, Microsoft, Ask.com και NASA. Ακόμα, ένας χρήστης μπορεί να κάνει αναζήτηση και ζουμ.



Μια bird's-eye άποψη από την κύρια οθόνη του Flash Earth. Προσέξτε τα controls στο πάνω αριστερό μέρος που επιτρέπουν στον χρήστη να επιλέξει την πηγή των δεδομένων χαρτογράφησης.



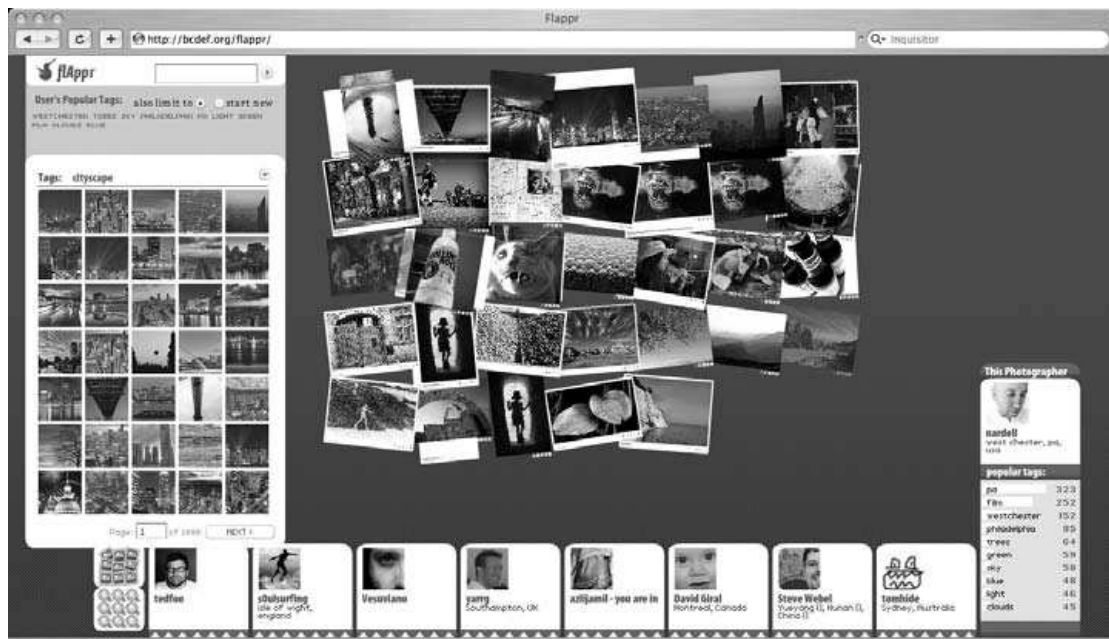
Μια μεγέθυνση του κεντρικού Brighton

2.1.2 Flappr (<http://bcdef.org/flappr/>)

Το Flappr χτίζεται επάνω στο Flickr API και αποτελεί ουσιαστικά μια εντελώς νέα διεπαφή που μπορεί να χρησιμοποιηθεί για την πρόσβαση στην photo-sharing ιστοσελίδα. Αυτό αποδεικνύει και πάλι τις εντυπωσιακές δυνατότητες του Flash, δίνοντας τη δυνατότητα στον χρήστη προβολής και περιήγησης μέσα σε μεγάλες συλλογές φωτογραφιών.



Εναλλακτική προβολή δεδομένων στο Flickr



Ξεφυλλίζοντας ένα σωρό φωτογραφίες

2.1.3 Διαδίδοντας μια διαδικτυακή εφαρμογή

Οι προγραμματιστές εφαρμογών web έχουν πολλές επιλογές όταν πρόκειται να δώσουν την εφαρμογή τους στους προγραμματιστές mashup. Συχνά, ένας από τους απλούστερους τρόπους είναι με τη δημοσίευση ως ένα RSS feed.

2.1.4 Feeds

Το RSS είναι ουσιαστικά ένα Extensible Markup Language (XML) έγγραφο το οποίο περιέχει λεπτομέρειες του εκδότη feed και οποιοδήποτε αριθμό στοιχείων που αντιστοιχούν σε δεδομένα της web εφαρμογής όπως ειδήσεις, φωτογραφίες, λίστες, και πολλά άλλα.

Το ακόλουθο είναι ένα παράδειγμα ένα RSS feed:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rss xmlns:media="http://search.yahoo.com/mrss/" version="2.0">
<channel>
<title>BBC News | Technology | UK Edition</title>
<link>
http://news.bbc.co.uk/go/rss/-/1/hi/technology/default.stm
</link>
<description>
Visit BBC News for up-to-the-minute news, breaking news,
video, audio and feature stories. BBC News provides
trusted World and UK news as well as local and regional
perspectives. Also entertainment, business, science,
technology and health news.
</description>
<language>en-gb</language>
<lastBuildDate>Mon, 25 Jun 2007 17:24:34 GMT</lastBuildDate>
<copyright>
Copyright: (C) British Broadcasting Corporation, see
http://news.bbc.co.uk/1/hi/help/rss/4498287.stm for terms
and conditions of reuse
</copyright>
<docs>http://www.bbc.co.uk/syndication/</docs>
<ttl>15</ttl>
<image>
<title>BBC News</title>
<url>
http://news.bbc.co.uk/nol/shared/img/bbc_news_120x60.gif
</url>
<link>
http://news.bbc.co.uk/go/rss/-/1/hi/technology/default.stm
</link>
</image>
<item>
```

<title>Social sites reveal class divide</title>
<description>
Social networking sites are proving a good guide to socio-economic divisions in the US, reveals research.
</description>
<link>
<http://news.bbc.co.uk/go/rss/-/1/hi/technology/6236628.stm>
</link>
<guid isPermaLink="false">
<http://news.bbc.co.uk/1/hi/technology/6236628.stm>
</guid>
<pubDate>Mon, 25 Jun 2007 11:42:15 GMT</pubDate>
<category>Technology</category>
</item>
<item>
<title>Leader of net piracy gang jailed</title>
<description>
A Briton who operated a net piracy gang out of his Australian home is jailed in the US for 51 months.
</description>
<link>
<http://news.bbc.co.uk/go/rss/-/1/hi/technology/6237610.stm>
</link>
<guid isPermaLink="false">
<http://news.bbc.co.uk/1/hi/technology/6237610.stm>
</guid>
<pubDate>Mon, 25 Jun 2007 13:56:19 GMT</pubDate>
<category>Technology</category>
</item>
<item>
<title>California landing for Atlantis</title>
<description>
US space shuttle Atlantis touches down in California, after bad weather prevents a Florida landing.
</description>
<link>
<http://news.bbc.co.uk/go/rss/-/1/hi/sci/tech/6232320.stm>
</link>
<guid isPermaLink="false">
<http://news.bbc.co.uk/1/hi/sci/tech/6232320.stm>
</guid>
<pubDate>Fri, 22 Jun 2007 20:56:50 GMT</pubDate>
<category>Science/Nature</category>
</item>
<item>
<title>Computers 'can raise attainment'</title>
<description>
A study says high levels of computer technology can improve school results, but the picture is mixed.
</description>

```

<link>
http://news.bbc.co.uk/go/rss/-/1/hi/education/6231704.stm
</link>
<guid isPermaLink="false">
http://news.bbc.co.uk/1/hi/education/6231704.stm
</guid>
<pubDate>Sun, 24 Jun 2007 07:08:51 GMT</pubDate>
<category>Education</category>
</item>
<item>
<title>Yahoo sees advertising shake-up</title>
<description>
Yahoo is overhauling the way it sells advertising in the US,
in an effort to help revive the business.
</description>
<link>
http://news.bbc.co.uk/go/rss/-/1/hi/business/6236102.stm
</link>
<guid isPermaLink="false">
http://news.bbc.co.uk/1/hi/business/6236102.stm
</guid>
<pubDate>Mon, 25 Jun 2007 06:45:30 GMT</pubDate>
<category>Business</category>
</item>
</channel>
</rss>

```

Αυτό είναι ένα RSS feed από την ιστοσελίδα του BBC News. Ο κόμβος κανάλι περιέχει πληροφορίες σχετικά με τον feed εκδότη, περιλαμβάνει το όνομα των feeds, την περιγραφή, την διεύθυνση URL και δημοσιεύθηκε σήμερα. Περιέχει επίσης πολλά <item> κόμβους που αντιστοιχούν σε μεμονωμένες ειδήσεις, το καθένα με το δικό του τίτλο, την περιγραφή του, URL, την κατηγορία, και την ημερομηνία δημοσίευσης. Τα RSS feeds χρησιμοποιούνται συχνά από aggregators-desktop ή web εφαρμογές που παρακολουθούν μια λίστα με feeds και τις παρουσιάζουν σε ένα μόνο interface. Desktop παραδείγματα περιλαμβάνουν Newsfire, Sage, και NetNewsWire. Τα online ισοδύναμά τους είναι το Google Reader και Jeremy Keith της Adactio Elsewhere. Επιπλέον, οι περισσότερες από τις μεγάλες μηχανές αναζήτησης στο διαδίκτυο υποστηρίζουν την ανάγνωση RSS feeds.

Για τους χρήστες υπάρχει ένα σαφές όφελος, αντί να επισκέπτονται πολλούς δικτυακούς τόπους, ώστε να μένουν ενημερωμένοι με τα γεγονότα της ημέρας, μπορούν απλά να εγγραφούν σε μια λίστα από feeds και να τα προβάλλουν σε ένα μόνο interface.

Η δημιουργία ενός αρχείου RSS είναι συνήθως μια διαδικασία που μπορεί να αυτοματοποιηθεί από μια web εφαρμογή. Για παράδειγμα, εάν φορτώσετε μια νέα φωτογραφία σε μια photo-sharing υπηρεσία όπως το Flickr, η εφαρμογή θα αποθηκεύσει τη φωτογραφία στη βάση δεδομένων της και την ίδια στιγμή θα κάνει update το home page και ένα RSS feed, πληροφορώντας για αυτήν την αλλαγή. Τότε ένα aggregator RSS θα εντοπίσει ότι ένα feed έχει ενημερωθεί και κατεβάζει την ενημέρωση.

Μπορείτε να ασφαλίσετε τα RSS feeds ακριβώς όπως σε μια ιστοσελίδα, έτσι για παράδειγμα, εάν ένα feed περιέχει ευαίσθητα δεδομένα, όπως το περιεχόμενο του inbox ενός χρήστη, τότε μπορείτε να το ρυθμίσετε να απαιτεί επικύρωση του χρήστη πριν από την προβολή του.

Τα Mashups βοηθούν τα feeds να περάσουν σε επόμενο επίπεδο, κάνοντας το format τους μια διεθνώς αναγνωρισμένη μορφή κωδικοποίησης δεδομένων. Αποτρέπουν πολύπλοκες και συχνά αναξιόπιστες τεχνικές, όπως η λήψη μιας ολόκληρης ιστοσελίδας και η προσπάθεια εξαγωγής συγκεκριμένων κομματιών δεδομένων. Αυτό είναι συχνά γνωστό ως screen scrap και είναι συχνά επιρρεπής σε σφάλματα, εάν το περιεχόμενο ή τη δομή μιας σελίδας αλλάξει, τότε η συλλογή δεδομένων μπορεί να αποτύχει. Τα RSS απλοποιούν σημαντικά τη χρήση (και την επαναχρησιμοποίηση) των δεδομένων μεταξύ των διαφόρων ιστοσελίδων και εφαρμογών.

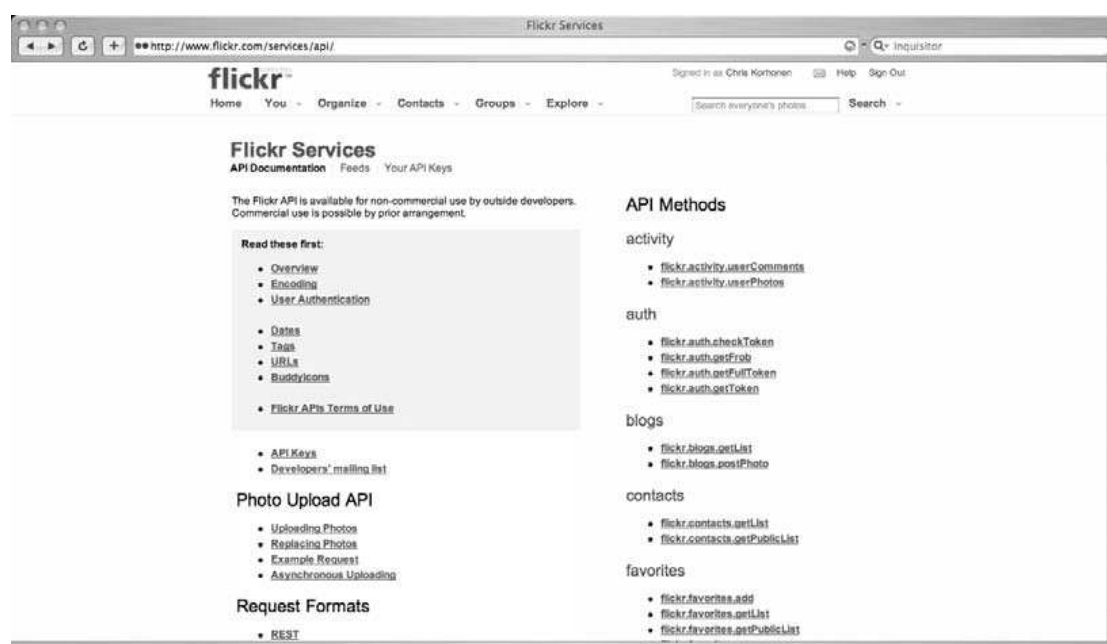
ΚΕΦΑΛΑΙΟ 3 Θεωρητικό Υπόβαθρο

3.1 APIs

Καλούμε **Διεπαφή Προγραμματισμού Εφαρμογών** (API, από το **Application Programming Interface**), γνωστή και ως **Διασύνδεση Προγραμματισμού Εφαρμογών** (για συντομία *διεπαφή* ή *διασύνδεση*), τη διεπαφή των προγραμματιστικών διαδικασιών που ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή παρέχει προκειμένου να επιτρέπει να γίνονται προς αυτό αιτήσεις από άλλα προγράμματα ή / και ανταλλαγή δεδομένων.

Ένας από τους βασικούς σκοπούς μίας διεπαφής είναι να ορίζει και να διατυπώνει το σύνολο των λειτουργιών-υπηρεσιών που μπορεί να παρέχει μια βιβλιοθήκη ή ένα λειτουργικό σύστημα σε άλλα προγράμματα, χωρίς να επιτρέπει πρόσβαση στον κώδικα που υλοποιεί αυτές τις υπηρεσίες. Η διεπαφή, ένα «συμβόλαιο κλήσης» μεταξύ καλούντος και καλούμενου, διαχωρίζει την προγραμματιστική υλοποίηση κάποιων υπηρεσιών από τη χρήση τους.

Π.χ. το ταχυδρομείο παρέχει την υπηρεσία της αποστολής γραμμάτων. Οι κανόνες οι οποίοι πρέπει να ακολουθηθούν για την υποβολή ενός αιτήματος αποστολής (φόρμα διεύθυνσης παραλαβής, γραμματόσημο κτλ) είναι καλώς ορισμένοι, αλλά το πώς θα υλοποιηθεί στην πράξη αυτό το αίτημα αφορά έναν ολόκληρο μηχανισμό υπαλλήλων εν πολλοίς αθέατο στον χρήστη της υπηρεσίας. Στο εν λόγω παράδειγμα διεπαφή είναι οι υπηρεσίες που παρέχονται στους πελάτες οι οποίες συνήθως είναι γραμμένες σε ένα φυλλάδιο, τη διεπαφή του ταχυδρομείου προς τους χρήστες του.



Εικόνα 1-12. Ορισμένες από τα πολλά API που είναι διαθέσιμα στους Flickr προγραμματιστές

3.2 Τα πλεονεκτήματα μιας open εφαρμογής

Πολλές εταιρίες και προγραμματιστές έχουν ανοίξει τις εφαρμογές τους στο διαδίκτυο, χρησιμοποιώντας ένα συνδυασμό των API και RSS feeds, προκειμένου να δημιουργηθεί μια πλατφόρμα πάνω στην οποία οι προγραμματιστές mashup να μπορούν να δουλέψουν. Ανάλογα με την εφαρμογή, οι λόγοι μπορεί να είναι πολύ διαφορετικοί.

3.2.1 Η καινοτομία ενθαρρύνεται

Το Google αναγνωρίζει ότι δεν κατέχει το μονοπώλιο της δημιουργικότητας. Πολλοί προγραμματιστές εκεί έξω αν τους δινόταν η ευκαιρία θα μπορούσαν να βρουν δεδομένα ή λειτουργίες που οι αρχικοί προγραμματιστές της εφαρμογής μπορεί να μην έχουν σκεφτεί. Ένα API δίνει στους προγραμματιστές τα εργαλεία για να πειραματιστούν και να δημιουργήσουν νέες εφαρμογές.

Σε ορισμένες περιπτώσεις, αυτή η δημιουργική παραγωγή μπορεί να παράγει νέες ιδέες και να επηρεάσει τη γενική κατεύθυνση του προϊόντος. Αυτές οι ιδέες μπορεί να είναι εντελώς αναπάντεχες και, τελικά, προσθέτουν σημαντική αξία στα δεδομένα.

Ως προγραμματιστής mashup, ένα μεγάλο μέρος της διασκέδασης είναι η δυνατότητα να πάρεις διαφορετικά σύνολα δεδομένων και να δημιουργήσεις νέα δεδομένα βλέποντας τα από άλλη οπτική γωνία. Αυτό θα μπορούσε να δώσει αξία. Για παράδειγμα, βάζοντας πληροφορίες σε ένα χάρτη, προσθέτετε υλικό το οποίο διαφορετικά θα λείπει. Εναλλακτικά, συνδυάζοντας τα δεδομένα παρέχεται μια μορφή δημιουργικής έκφρασης και ψυχαγωγίας.

3.2.2 Επέκταση της λειτουργικότητας

Ένα API σας επιτρέπει να επεκτείνετε τη λειτουργικότητα μιας web εφαρμογής, σε τομείς που δεν θα μπορούσαν να ενδιαφέρουν τους προγραμματιστές εφαρμογών, αλλά που συχνά είναι επιθυμητή από ένα υποσύνολο της βάσης χρηστών. Για παράδειγμα, Housing Maps (<http://www.housingmaps.com>) επεκτείνει τη λειτουργικότητα της Craigslist, επιτρέποντας μικρές αγγελίες να προβάλλονται σε ένα χάρτη.

3.2.3 Αύξηση της κυκλοφορίας

Ένα mashup μπορεί συχνά να είναι το μέσο που θα οδηγήσει ένα χρήστη σε μια ιστοσελίδα ή εφαρμογή. Μάλιστα, θα μπορούσε να οδηγήσει ένα χρήστη να εγγραφεί μια ιστοσελίδα και εν τέλει να αυξήσει την προβολή της.

Για παράδειγμα, τα πολλά mashups με Flickr feeds και APIs αποτελούν σημαντικό παράγοντα για την διαφήμιση του site που βρίσκονται. Τα Mashups βοηθούν στην φήμη των ιστοσελίδων στις online κοινότητες και χρησιμεύουν συχνά

για να ξεχωρίσουν από τους ανταγωνιστές τους.

Συχνά αυτό μπορεί να λειτουργήσει με δύο τρόπους, αν μια γνωστή εταιρία λανσάρει μια διεπαφή API για τις υπηρεσίες της, τότε οι προγραμματιστές θα είναι περισσότερο διατεθειμένοι να το ελέγξουν σε σύγκριση με μια άλλη νέα εταιρία. Ένα καλό παράδειγμα είναι η Amazon, και οι S3 υπηρεσίες της, καθώς είναι μια εταιρεία που φημίζεται για την αξιοπιστία της.

3.2.4 Κοινότητες

Ένα API μπορεί να είναι ένας τρόπος μέσω του οποίου θα αναπτυχθεί μια κοινότητα γύρω από μια συγκεκριμένη δικτυακή εφαρμογή. Αυτό μπορεί να είναι ζωτικής σημασίας για την επιτυχία μιας web εφαρμογής, διότι με άλλους προγραμματιστές να ασχολούνται με την εφαρμογή μπορούν να δημιουργηθούν τεράστια οφέλη σε όλους τους τομείς.

Οι social networking ιστοσελίδες όπως το Facebook, έχουν επιτυχία γιατί έχουν αναπτύξει τεράστιες κοινότητες προγραμματιστών που ενδιαφέρονται για την δημιουργία mashups και εφαρμογές στις πλατφόρμες τους. Αυτό έχει οδηγήσει σε μια έκρηξη λειτουργικότητας διαθέσιμη για τους χρήστες και παράγει τεράστια ποσά της δημοσιότητας.

3.2.5 Ανοικτά δεδομένα

Συχνά, τα feeds και τα APIs εξυπηρετούν έναν πρωταρχικό σκοπό, να επιτρέψουν την πρόσβαση στα δεδομένα που περιέχονται μέσα σε μια εφαρμογή. Αυτή η λειτουργικότητα είναι η αρχή των ανοικτών δεδομένων, δίνοντας στον χρήστη τον έλεγχο των δεδομένων που είναι αποθηκευμένα στην εφαρμογή. Αν θέλουν, οι χρήστες μπορούν να σταματήσουν να χρησιμοποιούν τις υπηρεσίες και να φύγουν παίρνοντας όλα τα δεδομένα μαζί τους.

Ακόμα, τα ανοικτά δεδομένα απλουστεύουν την επαναχρησιμοποίηση τους σε πολλές διαφορετικές εφαρμογές. Αν κοιτάξετε την πρόσφατη ανάπτυξη των social networking ιστοσελίδων, θα δείτε κάτι ενοχλητικό, ο χρήστης είναι αναγκασμένος για να διατηρήσει τα ίδια στοιχεία στο προφίλ του σε πολλές ιστοσελίδες (Facebook, MySpace, και ούτω καθεξής), να δαπανήσει επιπλέον χρόνο κάθε φορά που έχει ένα νέο αγαπημένο τραγούδι, για παράδειγμα!

Τα APIs παρέχουν λύση σε αυτό. Για παράδειγμα, η Facebook πλατφόρμα ανάπτυξης επιτρέπει εύκολη δημιουργία mashups και ενοποίηση με άλλες πηγές δεδομένων.

Στο τέλος, ναι, υπάρχει ο κίνδυνος οι χρήστες να είναι ίσως λιγότερο προσκολλημένοι σε μια εφαρμογή, αλλά σίγουρα αυτό θέτει μια πρόκληση για τους προγραμματιστές να αναπτύξουν τις εφαρμογές είναι τελικά αυτό που ζητάει ο χρήστης.

3.2.6 Εσωτερικά mashups

Τα Mashups, επίσης, δεν χρειάζεται να είναι public. Αντίθετα, μπορούν να χρησιμοποιηθούν για το όφελος της ίδιας της εταιρίας. Έτσι οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές μέσα από τις υποδομές της εταιρείας και επίσης να επαναχρησιμοποιήσουν απλά δεδομένα. Αυτό μπορεί να οδηγήσει σε σημαντική εξοικονόμηση χρημάτων.

Η Yahoo για παράδειγμα χρησιμοποιεί μια έννοια που ονομάζεται Hack Day. Με απλά λόγια, μια Hack Day είναι μια ημέρα όπου οι εργαζόμενοι αντιμετωπίζουν ως πρόκληση την ανάπτυξη μιας πρωτότυπης εφαρμογής και την παρουσιάζουν. Κάθε εφαρμογή πρέπει να χρησιμοποιήσει οποιοδήποτε αριθμό προϊόντων και υπηρεσιών της εταιρείας, feeds, και APIs.

Οι επιτυχείς εφαρμογές μπορούν να εξελιχθούν σε ολοκληρωμένα προϊόντα.

Αυτή η καινοτομία βοηθά τη δημιουργικότητα στην εταιρία και χρησιμεύει για την ενίσχυση του ηθικού των εργαζομένων δίνοντας τους την ευκαιρία να κάνουν κάτι διασκεδαστικό για μια ημέρα, εκτός των κανονικών καθηκόντων τους.

3.2.7 Moneymaking ευκαιρίες

Δεν υπάρχει τίποτα να λέει ότι ένα API θα πρέπει να είναι ελεύθερα διαθέσιμο σε όλους. Αν είστε ο ιδιοκτήτης δεδομένων που οι χρήστες είναι πρόθυμοι να πληρώσουν για την πρόσβαση τους σε αυτά, τότε δεν υπάρχει τίποτα που σας αναγκάζει να μην απαιτήσετε να πληρώσουν για τη χρήση του.

Στο Ηνωμένο Βασίλειο, εταιρείες όπως η Royal Mail το κάνουν αυτό για πολλά χρόνια. Αυτό επιτρέπει σε άλλες οργανώσεις να κάνουν αναζήτηση στη βάση δεδομένων για ένα μικρό ποσό. Η Amazon έχει επίσης εκδώσει ένα παρόμοιο μοντέλο για S3 (online file / data storage) υπηρεσία, τη χρέωση των χρηστών για χώρο στο δίσκο και εύρος ζώνης που χρησιμοποιούν.

3.3 Εισαγωγή στο Flex

Σε αντίθεση με το Flash που βασίζεται σε μπάρες στο πεδίο του χρόνου (timeline based) και απευθύνεται σε κατασκευαστές δυναμικής κίνησης εφαρμογών (animators / designers) το Flex είναι βασισμένο σε συστατικά / γεγονότα (component / event based) και απευθύνεται περισσότερο σε προγραμματιστές εφαρμογών και στο προγραμματιστικό κομμάτι δημιουργίας μιας εφαρμογής.

Πριν ξεκινήσουμε να μιλάμε για προηγμένες έννοιες κατά την κατασκευή Rich Flex-Based mashups, θα καλύψουμε μερικές από τις λειτουργίες που προσφέρονται και τα βήματα μέσω των οποίων μπορείτε να δημιουργήσετε μια απλή εφαρμογή χρησιμοποιώντας Flex 3 και το Flex Builder ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Από το Flex 2, η Adobe επικεντρώθηκε στην παροχή ενός περιβάλλοντος για την ανάπτυξη πλούσιων εφαρμογών στο Internet που τρέχει σε Flash Player, ακόμα και για προγραμματιστές που μπορεί να μην γνωρίζουν animation και γραφικά σε Flash.

Το Flex 3-framework, που χτίστηκε σε ActionScript 3.0, παρέχει κοινά στοιχεία και λειτουργικότητα που μπορεί να χρησιμοποιηθεί ως βάση για την ανάπτυξη εφαρμογών. Παρόλο που πολλές από τις λειτουργίες framework με την προηγούμενη έκδοση παραμένουν ίδιες, πολλά επιπλέον χαρακτηριστικά επεκτείνουν το σύνολο εργαλείων και κάνουν δυνατή την ανάπτυξη νέων τύπων εφαρμογών.

Μερικές από αυτές τις βελτιώσεις:

Υποστήριξη για τη δημιουργία AIR εφαρμογές, που μπορούν να τρέχουν στον υπολογιστή γραφείου

Ενισχυμένο skinning και styling, δίνοντας στον προγραμματιστή ακόμα μεγαλύτερο έλεγχο κατά την προσαρμογή των οπτικών στοιχείων της διεπαφής χρήστη. Το Flex 3 προσθέτει CSS προβολή σχεδίασης καθώς και στήριξη για οπτικά

στοιχεία, όπως η stateful component skins που καταργούν την ανάγκη δημιουργίας εικόνων ή SWFs για κάθε state.

Περισσότερα components, ειδικά για την εργασία με δεδομένα. Το DataGrid είναι

ενισχυμένο, και υπάρχει μια παραλλαγή που υποστηρίζει OLAP queries, τα οποία έχουν σχεδιαστεί για να καταστεί τον τεμαχισμό των δεδομένων γρήγορο και εύκολο. Αυτό με τη σειρά του ενισχύει σημαντικά την built-in δυνατότητα ανάλυσης των δεδομένων του Flex.

Developer εργαλεία, (performance, memory profilers) που σας επιτρέπουν να παρακολουθείτε την χρησιμοποίηση των πόρων του συστήματος των εφαρμογών σας για βελτιστοποίηση.

Runtime localization. Αντί να δημιουργήσει πολλαπλές version μιας εφαρμογής μπορείτε τώρα να χρησιμοποιήσετε το Resource Manager για τη διαχείριση των locales

Επιδόσεις βελτιώσεις, οι οποίες βελτιώνουν την ταχύτητα εκτέλεσης και μειώνουν τη χρήση μνήμης.

3.3.1 Χρησιμοποιώντας Design Mode

Το Design Mode είναι ένα εξαιρετικό mode για να ξεκινήσετε. Θα έχετε στη διάθεσή σας μια λίστα από visual components που μπορείτε απλά να σύρετε και να σχεδιάσετε την εφαρμογή σας. Ακόμα μπορείτε να τροποποιήσετε κάθε επιλογή για τα εν λόγω visual components. Στο κεφάλαιο αυτό, θα αποκτήσετε οικειότητα με τον τρόπο σχεδιασμού και θα αρχίσετε να σχεδιάζετε μια εφαρμογή οπτικά.

Canvas

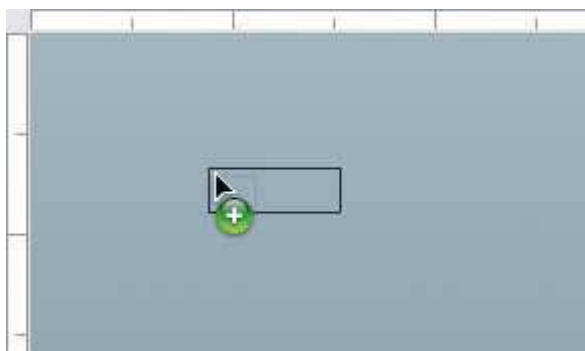
Εάν ανοίξετε μια κύρια εφαρμογή (ένα MXML αρχείο) σε Design Mode θα δείτε ένα κενό, μπλε φόντο, το προεπιλεγμένο στυλ για όλες τις Flex εφαρμογές. Όταν είστε σε Design Mode, το Components Panel θα πρέπει να είναι ορατό από προεπιλογή και να εμφανίζεται αριστερά κάτω δίπλα στο Outline Panel και κάτω από το Flex Navigator. Εάν το Components Panel είναι ορατό, μπορείτε να το εμφανίσετε από το μενού Window, το οποίο έχει τα πιο κοινά πάνελ για το Flex, δεξιά στην κορυφή του. Σημειώστε ότι το Components Panel δεν είναι διαθέσιμο όταν αλλάζετε σε Source Mode, επειδή σε Source mode θα χρησιμοποιήσετε MXML κώδικα για να δημιουργήσετε στοιχεία.

Μπορείτε να εμφανίσετε τον Components Panel μέσω Window → Components. Αυτό το Panel έχει όλα τα User Interface (UI), Components που θα σας βοηθήσουν για να σχεδιάσετε τα πάντα.

Προσθήκη στοιχείων στην Εφαρμογή

Για να τοποθετήσετε Components στην εφαρμογή σας, μπορείτε να τα μεταφέρετε από τον Components Panel και να τα τοποθετήσετε οπουδήποτε σας αρέσει στον καμβά σας, όπως φαίνεται στο παρακάτω σχήμα. Αυτά θα αποτελέσουν τον καμβά, και μπορείτε να τα μετακινήσετε και να τις ορίσετε τις ιδιότητες τους

οπτικά. Ας προσπαθήσουμε να τοποθετήσουμε ένα Button Component που θα βρείτε κοντά στην κορυφή του πίνακα Components, στο πλαίσιο Controls.



Μετακινώντας τα Components

Τώρα που έχετε ένα Button Components στη σκηνή, μπορείτε να το μετακινήσετε σύροντάς το σε διάφορα μέρη της σκηνής. Μπορείτε να παρατηρήσετε ότι αν το σύρετε προς τα άκρα της σκηνής, ορισμένες γραμμές αναφοράς εμφανίζονται για να σας βοηθήσουν να ευθυγραμμίσετε το στοιχείο. Αυτές είναι γνωστές ως οδηγοί ή οι έξυπνοι οδηγοί στο σχεδιαστικό κόσμο, και μπορεί να τις έχετε συναντήσει στο Photoshop ή στο αγαπημένο σας σχεδιαστικό πρόγραμμα. Οι οδηγοί αυτοί είναι μια οπτική αναπαράσταση του λεγόμενου snapping, που αναγκάζει το συστατικό να ευθυγραμμιστεί με τις ακμές ή με άλλα Components. Όταν σύρετε ένα στοιχείο σε μια άκρη της σκηνής, το snapping θα προκαλέσει κατά κανόνα ευθυγράμμιση του στοιχείου κατά 10 pixels από την άκρη. Όταν το σύρετε κοντά σε άλλα Components, οι οδηγοί εμφανίζονται και ευθυγραμμίζουν με απόσταση 10 ή 20 εικονοστοιχεία το ένα Component από το άλλο.

Εξερευνώντας κοινά Components

Ένα Component είναι απλά κάθε επαναχρησιμοποιήσιμο κομμάτι κώδικα που μπορεί να χρησιμοποιηθεί ως εργαλείο για να σχεδιάσετε την αίτησή σας. Ο όρος Component σημαίνει ότι θα πρέπει να χρησιμοποιείται ως ένα κομμάτι του συστήματος, και υπάρχουν περισσότερο από ένα εξ αυτών σε οποιοδήποτε εφαρμογή. Ένα από τα πιο χρήσιμα μέρη του Flex είναι η μεγάλη ποικιλία Components που προσφέρει. Και ενώ μπορούν να σας γλιτώσουν πολύ χρόνο και απαιτούν πολύ λίγη τεχνογνωσία, η άριστη γνώση τους μπορεί να πάρει λίγο χρόνο και μόλις την αποκτήσετε, είστε στο δρόμο να γίνετε ειδικός στο Flex.

Controls

Τα Controls είναι οπτικά Components που είναι βασικό μέρος του UI, όπως κουμπιά και το κείμενο. Ονομάζονται Controls, διότι ελέγχουν τη χρήση μιας εφαρμογής. Σκεφτείτε ένα τηλεχειριστήριο, το οποίο έχει κουμπιά για να επικοινωνεί με την τηλεόραση, ή με ένα stereo, το οποίο μπορεί να αλλάξει την ένταση του ήχου. Εδώ είναι μερικά από τα πιο κοινά Controls που είναι πιθανό να χρησιμοποιήσετε:

Button

Είναι ένα clickable Control που φαίνεται και λειτουργεί σαν ένα κουμπί. Έχει επίσης τη δυνατότητα toggle μέσω του toggle property, καθώς και η κατάσταση του μπορεί να διαφοροποιηθεί με το Selected Property.

CheckBox

Αυτό μοιάζει πολύ με ένα κουμπί, αλλά με το Toggle ως προεπιλογή.

ComboBox

Το ComboBox συνδυάζει ένα List Component με ένα Button Component, επιτρέποντας μια συμπαγή, drop-down/pop-up λίστα στοιχείων για επιλογή. Το selectedItem (αντικείμενο) ή selectedIndex (integer) property σας επιτρέπει να πάρετε ή να ορίσετε την επιλογή.

Image

Αυτό το Component σας επιτρέπει να φορτώσετε εικόνες. Οι πιθανοί τύποι είναι GIF, JPEG, PNG, και άλλα Swfs. Χρησιμοποιήστε το Source Property να ορίσετε το path που βρίσκεται η εικόνα.

Label

Αυτή είναι μια καλή λύση για ένα απλό, το αναγνωριστικό κείμενο μιας γραμμής.

List

Αυτό το στοιχείο παίρνει έναν κατάλογο αντικειμένων και τα εμφανίζει, επιτρέποντας την επιλογή και προσθέτοντας scroll bars εάν υπάρχουν πάρα πολλά στοιχεία για το μέγεθός της.

ProgressBar

Το Component αυτό είναι κατάλληλο για να τονίσουμε την πρόοδο των πραγμάτων όπως downloads. Εάν συνδυάσετε μια ProgressBar με ένα Image Component μπορείτε να δείτε την πρόοδο ενός download.

RadioButton

Μοιάζει με το CheckBox Control και χρησιμοποιείται σε ένα group που όταν επιλέξετε ένα, αποδιαλέγονται όλα τα άλλα. Ο ευκολότερος τρόπος να χρησιμοποιηθεί ο έλεγχος αυτός είναι σύροντας ένα RadioButtonGroup στη σκηνή, το οποίο θα εμφανίσει ένα παράθυρο διαλόγου που σας επιτρέπει να σχεδιάσετε ένα σύνολο κουμπιών οπτικά.

Text

Χρησιμοποιήστε το Text Component, όταν θέλετε να εμφανίσετε κομμάτια κειμένου που δεν θέλετε να έχουν κύλιση. Θα θέσει το μέγεθός του σε ένα τετράγωνο που να εφαρμόζει το κείμενο.

Textarea

Χρησιμοποιήστε ένα Textarea όταν θέλετε ένα κομμάτι του κειμένου που μπορεί να είναι μεγαλύτερο από τον διαθέσιμο χώρο. Θα προστεθεί κύλιση.

TextInput

Αυτό είναι ένα κοινό Component για είσοδο μιας γραμμής. Με αυτό, μπορείτε να πάρετε ή να θέσετε τις ιδιότητες ενός κειμένου (αυτό ισχύει για Text, Textarea, και RichTextEditor).

Layout Containers

Τα Layout Containers, ή απλά Containers είναι οπτικά στοιχεία που χρησιμοποιούνται για τη διαχείριση του τρόπου που η εφαρμογή είναι ευθυγραμμισμένη ή πώς οι έλεγχοι βρίσκονται σε σχέση μεταξύ τους. Μερικές φορές, θα έχουν μια σαφή οπτική αναπαράσταση, όπως ένα Panel's title bar.

Canvas

Αυτό είναι ένα Container που δεν έχει καμία ειδική διάταξη. Ορίστε το x και y για κάθε στοιχείο που τοποθετείτε μέσα για να δείξετε που θέλετε να εμφανιστεί.

Form

Αυτό το Container δημιουργεί μια HTML διάταξη. Εάν το χρησιμοποιήσετε μαζί με FormItem Components, θα τα στοιχίσει κάθετα, με τα πεδία της φόρμας στα αριστερά.

FormItem

Αυτό το Container παίρνει ένα Control όπως ένα TextInput και του δίνει τη δυνατότητα να έχει μια ετικέτα. Απλά ορίστε την ιδιότητα ετικέτα του δοχείου FormItem. Αν βάλετε μερικά από αυτά μέσα σε ένα Form Container, τα πάντα θα στοιχηθούν αυτόματα.

HBox

Το HBox παίρνει τα Components και τα στοιχίζει με οριζόντιο τρόπο.

Panel

Αυτό το στοιχείο είναι ένα Component που μοιάζει με ένα παράθυρο με μια γραμμή τίτλου. Όταν το τοποθετήσετε στη σκηνή σε Design Mode, η προεπιλεγμένη διάταξη θα είναι "absolute", που σημαίνει ότι τα Components που

τοποθετούνται στο εσωτερικό του θα πρέπει να έχουν συντεταγμένες x και y, σαν ένα καμβά. Αλλά μπορείτε να αλλάξετε τη διάταξη είτε σε «οριζόντια» είτε «κάθετη», καθώς, το οποίο σημαίνει ότι μπορείτε να ευθυγραμμίσετε controls με οποιονδήποτε τρόπο θέλετε.

VBox

Παίρνει τα Components και τα στοιχίζει κάθετα.

Navigators

Η δουλειά τους είναι να πάρουν ένα σύνολο από components κάνοντας μόνο ορισμένα ορατά κάθε φορά.

Accordion

Αυτό είναι παρόμοιο με το TabNavigator (εξηγείται στη συνέχεια), με εξαίρεση το ότι η στοίχιση γίνεται κάθετα.

TabNavigator

Παίρνει ένα σύνολο από containers, όπως HBoxes και Panels και εμφανίζει μόνο μια ομάδα κάθε φορά.

Υπάρχουν επίσης components που ονομάζονται non-visual τα οποία παρέχουν λειτουργικότητα που δεν είναι οπτική. Σκεφτείτε ένα στοιχείο δεδομένων που θα μπορούσε να κάνει τη σύνδεση με απομακρυσμένα δεδομένα πολύ εύκολη, αυτό θα ήταν ένα παράδειγμα ενός στοιχείου που δεν έχει πρόσωπο, σα να λέμε. Αυτό σημαίνει ότι για μη οπτικά components, δεν θα είστε σε θέση να χρησιμοποιήσετε Design τρόπο να τα προσθέσετε ή να τα τροποποιήσετε, αλλά θα πρέπει να γράψετε κώδικα.

Τροποποιώντας τα properties

Μπορείτε να τροποποιήσετε το button components σας απευθείας σε Design mode, όπως την αλλαγή της ετικέτας και την τροποποίηση του μεγέθους της. Μπορείτε να δοκιμάσετε την αλλαγή της ετικέτας του button component σας με διπλό κλικ στο κουμπί στη σκηνή. Θα εμφανιστεί μια εισαγωγή κειμένου που σας επιτρέπει να αλλάξετε την ετικέτα. Μπορείτε επίσης να σύρετε ένα από τα handles που εμφανίζονται γύρω από το component σας για να αλλάξετε το μέγεθος. Όσο για την μπλε κουκκίδες που περιβάλλουν το components σας, αυτά είναι τα handles που σας επιτρέπουν να ρυθμίσετε εύκολα τη δημιουργία περιορισμών που βασίζονται στην διάταξη.

To Properties Panel

Για να τροποποιήσετε τις ιδιότητες ενός στοιχείου, θα πρέπει να μάθετε άλλο ένα Panel, το Flex Properties Panel. Θα το δείτε κατά πάσα πιθανότητα στα δεξιά εκτός αν το έχετε μετακινήσει ή το έχετε κλείσει. Αν δεν το βλέπετε, μπορείτε να το εμφανίσετε επιλέγοντας Window → Flex Properties. Σημειώστε ότι αυτή η ομάδα δεν

εμφανίζεται στο Source mode, επειδή υποτίθεται δεν χρειάζεται, γιατί προγραμματίζετε με το χέρι έτσι κι αλλιώς. Αλλά στον τρόπο σχεδίασης μπορεί να είναι ο καλύτερος φίλος σας. Αλλάζει τις επιλογές του, όταν επιλέξετε διαφορετικά στοιχεία πάνω στη σκηνή, οπότε αν έχετε πατήστε ένα button component, κάνοντας κλικ επάνω στη σκηνή, θα παρατηρήσετε ότι το Flex Properties Panel αλλάζει και δείχνει το mx: Button στον τίτλο και μια σειρά από ιδιότητες για το κουμπί.

Standard View

Η Κανονική προβολή είναι η προεπιλεγμένη προβολή του πίνακα Flex Properties και περιλαμβάνει μια προσαρμοσμένη διάταξη για κάθε component. Είναι πολύ χρήσιμη για να τροποποιεί τις πιο κοινές ιδιότητες ενός component, όπως η ετικέτα του, το ID, το πλάτος και το ύψος. Είναι εξαιρετικά χρήσιμη για την τροποποίηση του στυλ του στοιχείου σας όπως η γραμματοσειρά, ο τύπος κειμένου (έντονη γραφή, πλάγια γραφή, με υπογράμμιση, και ούτω καθεξής), και της διαφάνειας.

Category View

Η Category View είναι μια άλλη άποψη του Panel Properties. Η άποψη αυτή οργανώνει το σύνολο των ιδιοτήτων ενός στοιχείου του σε μια ιεραρχική προβολή δέντρου. Αυτό μοιάζει πολύ με το πώς η Standard View είναι οργανωμένη σε ομάδες όπως η Common Layout και Style. Ωστόσο, κατά την άποψη της Category View έχετε κάθε δυνατή ιδιότητα διαθέσιμη.

Alphabetical View

Η τρίτη και τελευταία διαθέσιμη προβολή Panel Properties είναι η Alphabetical View. Η Alphabetical View, έχει όλα τα πιθανά διαθέσιμα properties. Εντούτοις, εκτός από την κατηγορία Standard View, καμία δεν είναι οργανωμένοι σε αλφαβητική σειρά. Είναι απλά μια λίστα που οργανώνεται από το Α έως το Ω. Αυτό είναι χρήσιμο όταν γνωρίζετε το όνομα ή τουλάχιστον τα πρώτα γράμματα της ιδιότητας που θέλετε να τροποποιήσετε. Είναι επίσης χρήσιμο για να ρίξετε μια ματιά στις ιδιότητες που έχουν τεθεί σε ένα συγκεκριμένο components. Μπορείτε να σύρετε μερικά στοιχεία στη σκηνή και να τα μετακινήσετε όπου θέλετε. Τροποποιήσει ορισμένες ιδιότητες όπως η ετικέτα, το πλάτος, και το χρώμα για να πάρετε μια ιδέα για το πώς λειτουργεί.

Common Properties

Όλα τα visual components Flex (αυτά που μπορείτε να χειριστείτε σε κατάσταση σχεδίασης) κληρονομούν από μια βάση ένα component που ονομάζεται UIComponent. Αυτό το στοιχείο έχει πολλές ιδιότητες που σχετίζονται με την προβολή, το ποντίκι και το πληκτρολόγιο την αλληλεπίδραση, την εστίαση, το μέγεθος, καθώς και όλα τα καλούδια που θέλετε να προσθέσετε. Επειδή τα κουμπιά, text inputs, και τα containers επεκτείνουν αυτό το component, που σημαίνει ότι θα κληρονομήσει όλες τις λειτουργίες. Μόλις μάθετε την χρήση των properties για ένα common component, θα είστε σε θέση να την χρησιμοποιήσετε και σε άλλα components. Πάρτε το width property ενός button control, για παράδειγμα. Επειδή το πλάτος είναι μία ιδιότητα των εν λόγω component που μοιράζεται μεταξύ των visual

components, αν κατανοήσετε πώς να το χρησιμοποιήσετε με ένα κουμπί θα σας βοηθήσει να το χρησιμοποιήσετε και με άλλα components.

Ακολουθεί μια λίστα με τις πιο συχνά χρησιμοποιούμενες ιδιότητες για τα πιο κοινά components, μαζί με ένα παράδειγμα για κάθε ένα:

ID

Το id (συντόμευση του identifier) είναι μια πολύ σημαντική ιδιότητα, γιατί είναι το όνομα που δίνετε σε ένα component. Εάν έχετε τοποθετήσει δύο κουμπιά πάνω στη σκηνή, καλώντας τα Button1 και button2 σας επιτρέπει να διακρίνετε αργότερα στον κώδικα. Ακόμα καλύτερα είναι όταν τους δίνετε περιγραφικά ονόματα που βασίζονται σε ό, τι κάνουν στην αίτηση (για παράδειγμα, submitButton και refreshButton). Δεν είναι πάντα απαραίτητο να πληκτρολογήσετε ένα αναγνωριστικό, διότι το Flex εκχωρεί ένα από μόνο του. Αλλά είναι χρήσιμο να το κάνετε.

```
<mx:Button id="submitButton"/>
```

x

Αυτό είναι ένας αριθμός που παίρνει ή ορίζει τον αριθμό των pixels ενός στοιχείου που πρέπει να εμφανίζεται στα δεξιά. Αυτό σημαίνει ότι, όταν ένα στοιχείο τοποθετείται μέσα σε ένα container, και ότι το κουμπί του x τιμή έχει οριστεί σε 20, το κουμπί θα εμφανίσει 20 εικονοστοιχείων προς τα δεξιά από την άκρη του component.

```
<mx:Button x="20"/>
```

y

Αυτό είναι ένας αριθμός που παίρνει ή ορίζει τον αριθμό των pixels ενός στοιχείου πρέπει να εμφανίζεται από την κορυφή. Παρόμοια με το x, εκτός από το ότι αυτή η ιδιότητα ελέγχει την κατακόρυφη θέση αντί οριζόντια θέση.

```
<mx:Button y="10"/>
```

visible

Αυτή η ιδιότητα ελέγχει αν ένα στοιχείο θα είναι ορατό στη σκηνή. Ορίζοντας την ρύθμιση visible σε false θα κάνει ένα στοιχείο να εξαφανιστεί από την προβολή, αλλά θα εξακολουθεί να καταλαμβάνει χώρο. Για παράδειγμα, αν έχω τέσσερα κουμπιά μέσα σε ένα δοχείο HBox, ευθυγραμμίζοντας και τα τέσσερα οριζόντια, και κάνω το δεύτερο κουμπί άορατο, τα δύο κουμπιά στα δεξιά του και το ένα στα αριστερά του θα παραμείνει στον ίδιο σημείο, επειδή το άορατο κουμπί καταλαμβάνει ακόμα χώρο.

```
<mx:HBox>  
  <mx:Button label="One"/>  
  <mx:Button label="Two" visible="false"/>
```

```

    <mx:Button label="Three"/>
    <mx:Button label="Four"/>
</mx:HBox>

```



includeInLayout

Η Ρύθμιση αυτή σε false για ένα component θα προκαλέσει ένα container να αγνοήσει αυτό το component κατά την διάταξη. Αν δεν θέλετε κάτι να καταλαμβάνει χώρο, μπορείτε να το ορίσετε σε false. Είναι χρήσιμο όταν θέλετε να κάνετε ένα component αόρατο χωρίς να επηρεαστεί η διάταξή σας.

```

<mx:HBox>
  <mx:Button label="One"/>
  <mx:Button label="Two" visible="false" includeInLayout="false"/>
  <mx:Button label="Three"/>
  <mx:Button label="Four"/>
</mx:HBox>

```



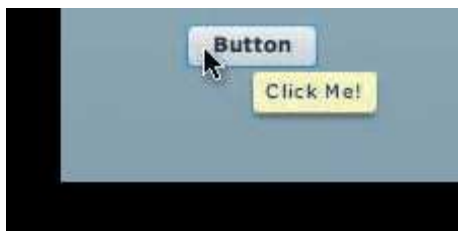
tooltip

Ένα tooltip είναι ένα μικρό αναδύμενο μήνυμα που εμφανίζεται όταν τοποθετείτε το ποντίκι σας πάνω από ένα στοιχείο για μια στιγμή. Μπορεί να βοηθήσουν στην περιγραφή του στοιχείου. Όλα τα Flex UI components υποστηρίζουν τα tooltip, οπότε είναι εύκολο να εφαρμοστούν.

```

<mx:Button tooltip="Click Me!"/>

```

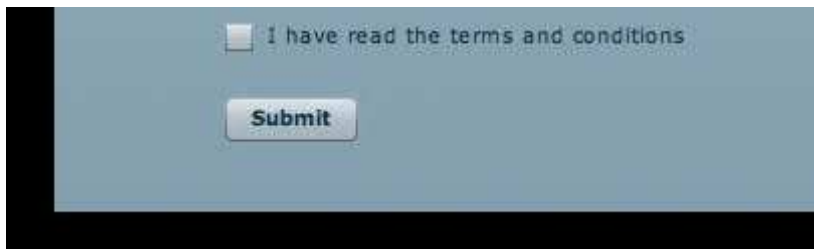


Label

Δεν χρησιμοποιούν όλα τα components labels, αλλά πολλά. Για ένα κουμπί, ένα label είναι το κείμενο που εμφανίζεται στο κουμπί. Στα Checkbox χρησιμοποιείται με παρόμοιο τρόπο. Στα containers χρησιμοποιείται επίσης. Για παράδειγμα, το FormItem container εμφανίζει το label, και μπορεί να χρησιμοποιηθεί ως περιγραφή ενός πεδίου, όταν συνδεθεί με ένα άλλο στοιχείο ελέγχου, όπως ένα TextInput, σε ένα Form. Για τα άλλα container διάταξης, η ετικέτα θα εμφανιστεί

όταν συνδυάζεται με στοιχεία πλοήγησης. Δηλαδή, όταν χρησιμοποιείται με ένα TabNavigator, η ετικέτα του container θα εμφανιστεί ως η ετικέτα της καρτέλας που επιλέγεται.

```
<mx:CheckBox label="I have read the terms and conditions"/>  
<mx:Button label="Submit"/>
```



text

Αυτή η ιδιότητα αλλάζει το κείμενο που εμφανίζεται στο text controls, όπως TextInput, Textarea, και Label.

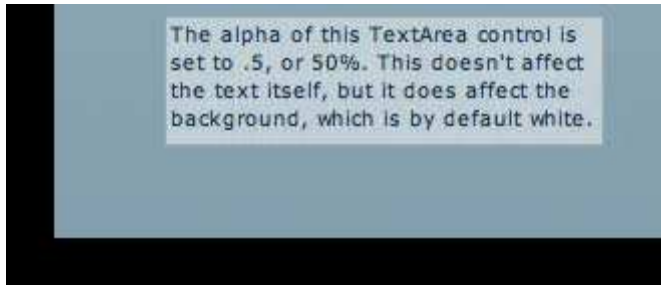
```
<mx:Label text="Enter Your Name:"/>  
<mx:TextInput text="Alaric"/>
```



Alpha

Το Alpha είναι συντόμευση για το alpha channel, και είναι ένας αριθμός από 0 έως 1 που ελέγχει πόσο διαφανής είναι ένα στοιχείο. Αν θέλετε, να εμφανίζεται τι είναι πίσω από ένα component μπροστά, μπορείτε να ορίσετε την τιμή άλφα σε λιγότερο από 1. Είναι percentagebased, οπότε 0,9 σημαίνει 90 τοις εκατό αδιαφανές, 0,45 σημαίνει 45 αδιαφανές τοις εκατό, και ούτω καθεξής. Σε περίπτωση ελέγχου καταλόγου, θέτοντας άλφα σε .5 θα επιτρέψει το έγχρωμο φόντο της εφαρμογής σας για να εμφανιστεί μπροστά.

```
<mx:TextArea alpha="0.5" text="The alpha of this TextArea control is  
set to .5"/>
```

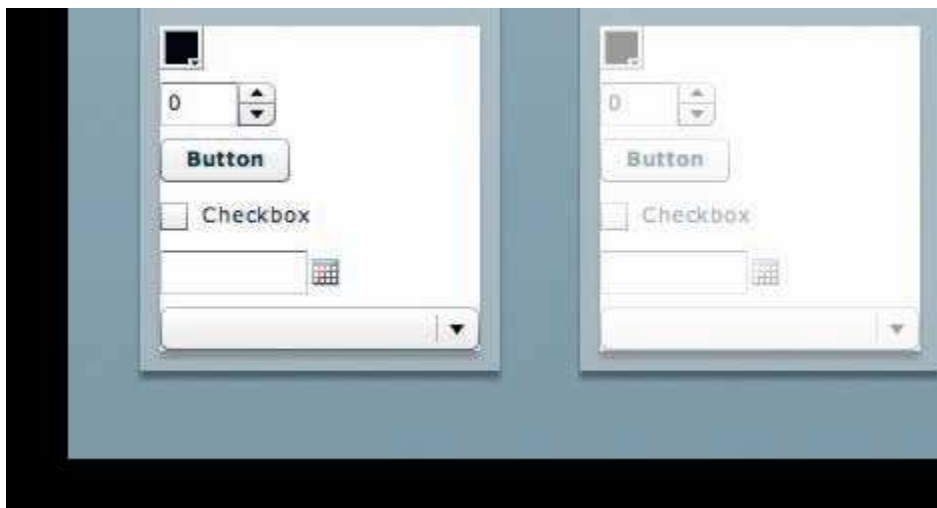


enabled

Ρυθμίζοντας το enable σε false κάνει ένα component μη διαθέσιμο. Συνήθως, αυτό κάνει το component γκρι και το κάνει ακόμα να μην ανταποκρίνεται στα κλικ από το ποντίκι όπως φαίνεται στην εικόνα. Αυτό είναι χρήσιμο αν θέλετε να απενεργοποιήσετε ένα κουμπί που δεν πρέπει να χρησιμοποιείται ακόμα, αλλά λειτουργεί επίσης με containers. Μπορείτε να ορίσετε τη ιδιότητα του Panel false και όλα μέσα σε αυτό θα απενεργοποιηθούν επίσης.

```
<mx:Panel title="Enabled Panel" enabled="true">
  <mx:ColorPicker/>
  <mx:NumericStepper/>
  <mx:Button label="Button"/>
  <mx:CheckBox label="Checkbox"/>
  <mx:DateField/>
  <mx:ComboBox/>
</mx:Panel>
```

```
<mx:Panel title="Disabled Panel" enabled="false">
  <mx:ColorPicker/>
  <mx:NumericStepper/>
  <mx:Button label="Button"/>
  <mx:CheckBox label="Checkbox"/>
  <mx:DateField/>
  <mx:ComboBox/>
</mx:Panel>
```

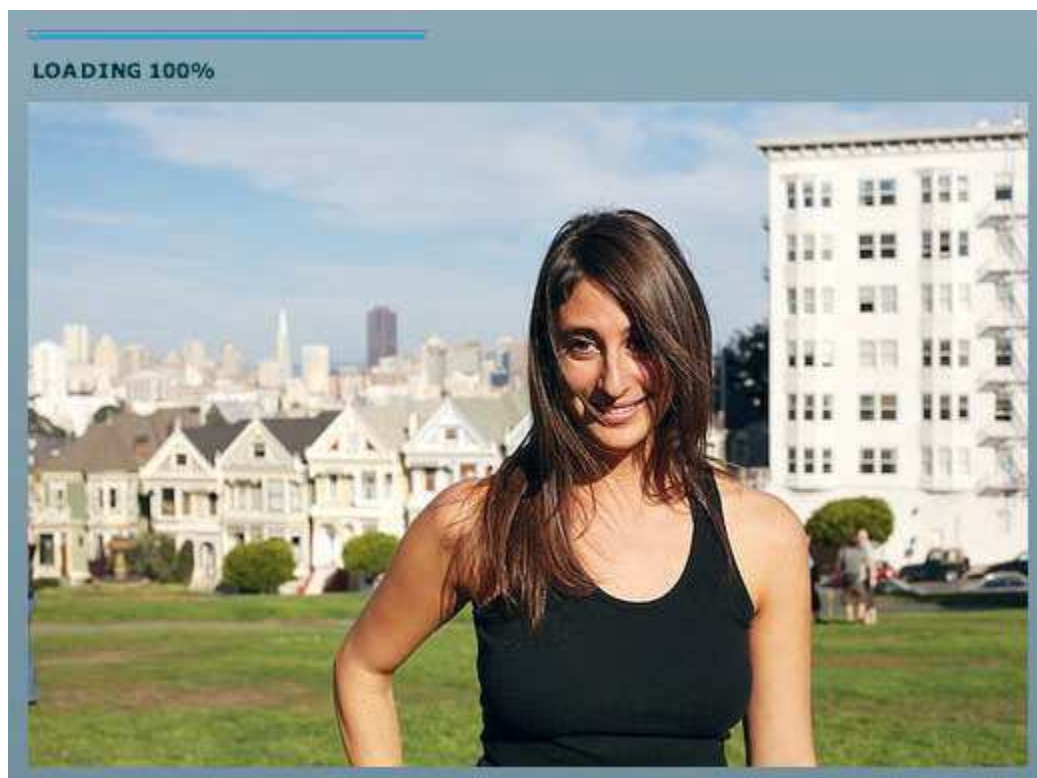


Source

Μπορείτε να χρησιμοποιήσετε αυτήν την ιδιότητα για τους ελέγχους εικόνας και άλλους έλεγχοι από εξωτερικά αρχεία. Μπορείτε επίσης να την χρησιμοποιήσετε με τον έλεγχο ProgressBar και να καθορίσετε την πρόοδο αυτού του ελέγχου εικόνας. Για παράδειγμα, δύο χρήσεις της ιδιότητας θα είναι μια για το τι στοιχείο ελέγχου εικόνας θα πρέπει να φέρει και μια για το τι θα πρέπει να δείχνει η ProgressBar.

```
<mx:ProgressBar source="{photo}"/>
```

```
<mx:Image id="photo" source="http://greenlike.com/photos/lydia.jpg" />
```



Η κατασκευή μιας διεπαφής χρήστη

Ας κατασκευάσουμε ένα απλό παράδειγμα χρησιμοποιώντας μερικά κοινά στοιχεία:

1. Σύρετε ένα πάνελ μέσα στο stage. Το Panel είναι ένας τύπος container και θα το βρείτε στον πίνακα Components Panel στην ενότητα "Layout".
2. Δώστε στο Panel έναν τίτλο κάνοντας διπλό κλικ ή μπαίνοντας στις ιδιότητες Panel. Θα χρησιμοποιήσω τη φράση Howdy Ya'll
3. Σύρετε ένα στοιχείο Label στο Panel. Επειδή το Panel είναι ένα Container, μπορεί να κρατήσει τα στοιχεία ακριβώς όπως και το component Application το οποίο είναι επίσης ένα container. Έτσι, αν κάνετε drop ένα Label σε αυτό το πάνελ, τοποθετήστε το κοντά στην

κορυφή του panel.

4. Δώστε στην Ετικέτα του κειμένου “My name is”:, και κάντε το κείμενο bold. Μπορείτε να το κάνετε bold επιλέγοντας την ετικέτα και κάνοντας κλικ στο bold icon στο Flex Properties

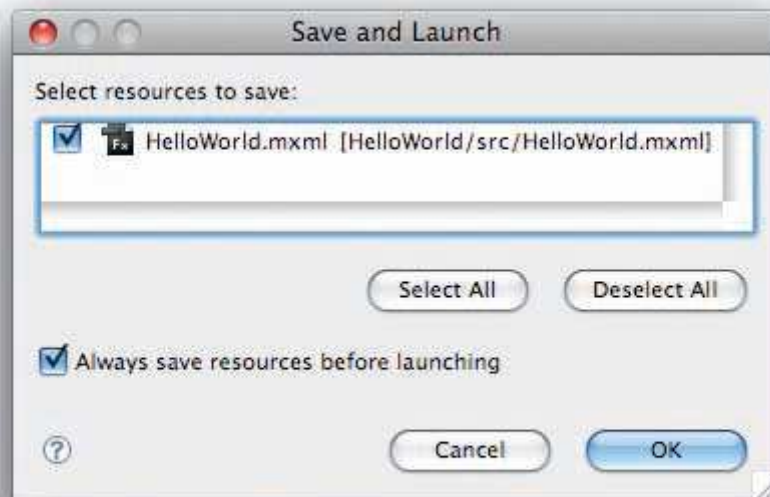
panel στην ενότητα Style.

5. Κάντε Drop ένα TextInput control κάτω από το Label.

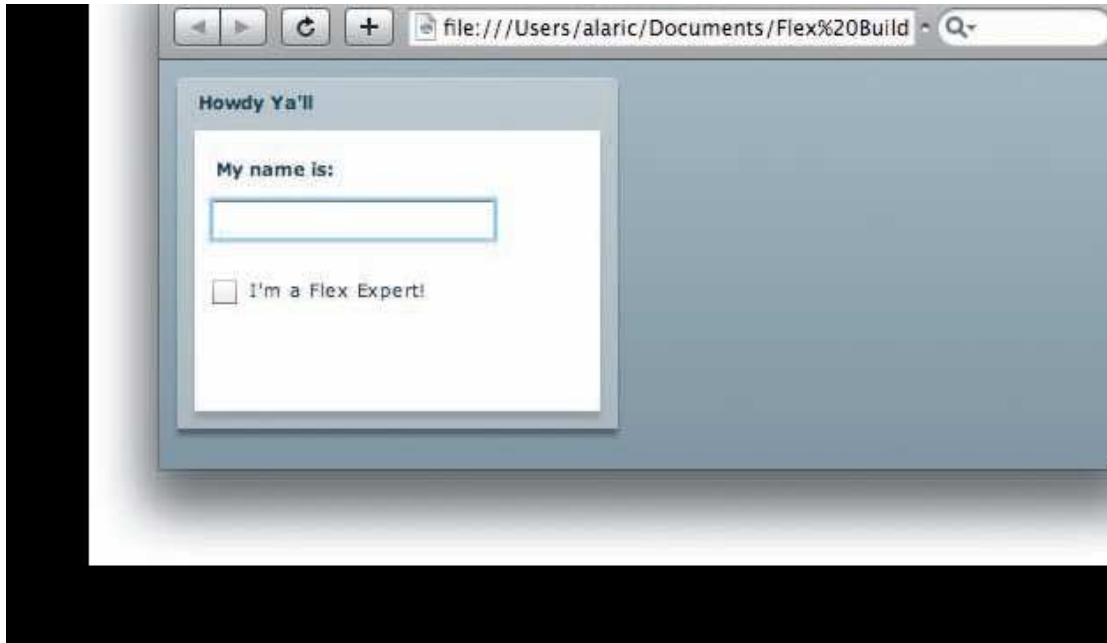
6. Κάντε Drop ένα CheckBox control κάτω από το TextInput.

7. Δώστε στο CheckBox το label I’m a Flex Expert!. (Μπορείτε να τροποποιήσει την ετικέτα με τον ίδιο τρόπο όπως και στα περισσότερα components, κάνοντας διπλό κλικ πάνω στη σκηνή.)

Έχετε δημιουργήσει το user interface για την πρώτη Flex εφαρμογή σας. Εάν το τρέξετε σε ένα πρόγραμμα περιήγησης, κάνοντας κλικ στο κουμπί "Run" θα δείτε τα ακόλουθα:



Θα εμφανιστεί η εφαρμογή σας σε ένα παράθυρο του browser. Μπορείτε να πληκτρολογήσετε το όνομά σας στο text input και ενεργοποιήστε το checkbox που λέει "I’m Flex Expert"



Τώρα είστε εξοικειωμένοι με τρόπο σχεδιασμού, είστε σε θέση να δημιουργήσετε μια εφαρμογή τα standard Flex components. Έχετε ακόμα την δυνατότητα να μάθετε πως να κάνετε τις εφαρμογές σας διαδραστικές.

Χρησιμοποιώντας SOURCE MODE

Αυτό που κάνει το Design mode είναι να δημιουργεί MXML κώδικα για σας. Όταν σύρετε ένα component στη σκηνή, το Design mode γράφει ένα αντίστοιχο Label στην κύρια εφαρμογή του .MXML αρχείου. Εάν τροποποιήσετε τον MXML κώδικα σε Source Mode, ενημερώνεται το Design Mode και δείχνει αυτές τις αλλαγές. Ανοίξτε την εφαρμογή που ξεκινήσατε στο προηγούμενο κεφάλαιο και στη συνέχεια, μεταβείτε στο Source Mode για να δούμε τον κώδικα που δημιούργησε για εσάς.

Η Ανατομία μιας Flex Αίτησης

Κοιτάζοντας τον πηγαίο κώδικα για αυτό το απλό παράδειγμα HelloWorld, θα δείτε κάτι σαν το παρακάτω:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute">
  <mx:Panel x="10" y="10" layout="absolute" title="Howdy Ya'll">
    <mx:Label text="My name is:" fontWeight="bold" x="10" y="14"/>
    <mx:TextInput x="5" y="41"/>
    <mx:CheckBox label="I'm a Flex Expert!" x="10" y="71"/>
  </mx:Panel>
</mx:Application>
```

Τώρα θα εξηγήσουμε αυτόν τον κώδικα κάτω για σας. Η πρώτη γραμμή περιέχει μια δήλωση XML, η οποία είναι προαιρετική. Στο επόμενο με την ετικέτα

Application αρχίζει το πρόγραμμα.

Κάθε εφαρμογή Flex αρχίζει με ένα root tag. Για Flex εφαρμογές που έχουν αναπτυχθεί στο διαδίκτυο, το root tag ονομάζεται Application. Για τις AIR εφαρμογές,

το root tag ονομάζεται WindowedApplication και είναι λίγο πολύ ίδιο, με μερικές μόνο desktop-ιδιαιτερότητες. Για την εφαρμογή σας, η οποία πρόκειται να προβληθεί σε πρόγραμμα περιήγησης, θα δείτε να εμφανίζεται το root tag `<mx:Application>`. Παρατηρήστε την ιδιότητα layout "Absolute". Υπάρχει από προεπιλογή στο Flex και σημαίνει: "να τοποθετηθούν τα πάντα εδώ στις x, y συντεταγμένες." Λόγω του ότι μια εφαρμογή είναι ένα container, έχει την δυνατότητα, είτε να ζητήσει x, y συντεταγμένες ή να τοποθετηθούν όλα σε στοίβες ή γραμμές.

Το Panel μέσα στο Application Tag είναι επίσης ένα container και έχει προεπιλεγμένη διάταξη "absolute" επίσης. Προσέξτε τις ιδιότητες `x = "10"` και `y = "10"`, που λένε στο Flex να τοποθετήσει στον πίνακα 10 pixels από την αριστερή γωνία της εφαρμογής. Τα Label, TextInput, και Checkbox έχουν επίσης συντεταγμένες, αλλά αυτές οι συντεταγμένες είναι σχετικές με τον container panel στο οποίο εμφανίζονται.

Φυσικά, αυτές οι ετικέτες να έχουν άλλα χαρακτηριστικά. Ρίξτε μια ματιά στο panel, του οποίου ο τίτλος έχει οριστεί μέσω της title ιδιότητας. Επιπλέον, η ετικέτα χρησιμοποιεί το text property για να ορίσει το κείμενο και έχει ένα style property που ονομάζεται fontWeight για να γίνει το κείμενο bold. Τέλος, το Checkbox χρησιμοποιεί μια ιδιότητα που ονομάζεται label για να ορίσετε την ετικέτα του.

Δοκιμάστε να επεξεργαστείτε τον κώδικα και να αλλάξετε ορισμένα από τα properties. ακίνητα. Θα μπορούσατε, για παράδειγμα, να αλλάξετε την ετικέτα του Checkbox να λέει "I'm a Flex Genius" Όταν τελειώσετε, μεταβείτε σε Design mode για να δείτε τις αλλαγές.

Components σε Source Mode

Τώρα δοκιμάστε την τροποποίηση του UI, πληκτρολογώντας την MXML για να προσθέσετε ένα component. Δεξιά κάτω από το Checkbox μέσα στο panel, προσθέστε ένα button component πληκτρολογώντας `<mx:Button label="Click me"/>`.

Εάν πάτε πίσω σε Design Mode, μπορείτε να παρατηρήσετε ότι το κουμπί τοποθετείται πάνω από οτιδήποτε άλλο και ίσως και ίσως όχι εκεί που θέλατε (κάτω από το Checkbox). Αυτό συμβαίνει γιατί δεν έχετε ρυθμίσει τις x, y συντεταγμένες του ακόμη και το Flex έχει προεπιλογές για `x = "0"` και `y = "0"`. Τώρα, έχετε την επιλογή να μαντέψετε τις συντεταγμένες και να τις πληκτρολογήσετε σε Source mode ή να μετακινήσετε το κουμπί σε Design mode. Στην πραγματικότητα, έχετε μια άλλη επιλογή. Αν θέλετε μια κάθετη εμφάνιση για τα αντικείμενα στο panel σας αντί να ρυθμίσετε σε κάθε στοιχείο τις x, y συντεταγμένες, θα μπορούσατε να τις αλλάξετε από το properties panel.

Τώρα τα items στοιβάζονται το ένα πάνω στο άλλο κάθετα, και αυτό είναι υπέροχο. Αλλά είναι κολλημένα στο αριστερό άκρο του panel με ένα όχι και τόσο ελκυστικό τρόπο. Σε αυτό το σημείο χρησιμοποιούμε την paddingleft property. Εάν ορίσετε την paddingleft property της ομάδας σε 5, θα δημιουργήσει ένα χώρο μεταξύ στην αριστερή πλευρά της ομάδας και το περιεχόμενό της.

Συμπλήρωση κώδικα

Έχετε δει χωρίς αμφιβολία ότι το Flex Builder αναδύει μια λίστα αυτόματης συμπλήρωσης καταχωρήσεων όταν αρχίσετε να πληκτρολογήσετε κάτι στο πρόγραμμα επεξεργασίας σε λειτουργία Source. Αυτή η αυτόματη συμπλήρωση είναι ένα από τα πιο χρήσιμα και ισχυρά χαρακτηριστικά γνωρίσματα του. Τυπογραφικά και ορθογραφικά λάθη μπορούν να καταστρέψουν το έργο σας. Όταν αρχίσετε να πληκτρολογείτε μια ιδιότητα, το Flex Builder επεμβαίνει επάνω τις προτάσεις, και σας προτείνει μια πιθανή λύση, έτσι πατώντας Enter εισάγει το πλήρες κείμενο για σας. Αυτή η αυτόματη συμπλήρωση είναι χρήσιμη όχι μόνο για τις ιδιότητες αλλά και για λέξεις-κλειδιά. Αν είστε έτοιμος για να εισαγάγετε ένα κουμπί, δεν χρειάζεται καν να ξεκινήσετε τη σύνταξη, γράφοντας το mx: κομμάτι μόνο εμφανίζεται η λίστα και επιλέγετε ότι εσείς θέλετε.



Η Ανατομία ενός Tag

Ένα tag μπορεί να περιέχει πληροφορίες με δύο τρόπους, είτε με περιεχόμενο ή κατά ιδιότητες. Το περιεχόμενο είναι απλώς το κείμενο που υπάρχει μεταξύ δύο ετικετών, και τα χαρακτηριστικά είναι το κείμενο μέσα στην ετικέτα ανοίγματος μόνο, με τις πληροφορίες να περικλείονται σε εισαγωγικά. Κοιτάξτε τον ακόλουθο κώδικα XML:

```

<book title="Learning Flex" author="Alaric Cole">
  <chapter title="Getting Up to Speed"/>
  <chapter title="Setting Up Your Environment"/>
</book>

```

Σε αυτό το παράδειγμα, το `<book>` είναι το root tag, καθώς ο τίτλος και ο συγγραφέας του εν λόγω tag τοποθετούνται ως χαρακτηριστικά του εν λόγω βιβλίου. Ο κώδικας έχει επίσης δύο ετικέτες που αντιπροσωπεύουν τα δύο κεφάλαια. Συγκρίνετε τον προηγούμενο κώδικα με τον ακόλουθο, ο οποίος περιέχει τις ίδιες πληροφορίες, αλλά τοποθετημένες με ένα διαφορετικό τρόπο:

```

<book>
  <title>Learning Flex</title>
  <author>Alaric Cole</author>

```

```

<chapter>
  <title>Getting Up to Speed</title>
</chapter>
<chapter>
  <title>Setting Up your Environment</title>
</chapter>
</book>

```

Ο κώδικας αυτός είναι ουσιαστικά ο ίδιος με τον πρώτο, αλλά θα παρατηρήσετε ότι είναι λίγο πιο αναλυτικός. Ο πρώτος χρησιμοποιεί attributes, ενώ ο δεύτερος χρησιμοποιεί tags. Έτσι, τα attributes μπορεί να είναι χρησιμοποιηθούν ως ένας πιο συμπαγής τρόπος να αναπαρασταθούν οι ίδιες πληροφορίες με τα tags. Συγκρίνετε αυτό τον κωδικό για το ίδιο παράδειγμα σε MXML.

Ωστόσο, μερικές φορές θα χρειαστεί να tags για τα properties αντί για τα attributes. Ας εξετάσουμε την περίπτωση του text property. Εάν το κείμενο ήταν μια ολόκληρη παράγραφο, θα φαινόταν παράξενο να τοποθετήσουμε το inline σαν ένα attribute. Ακόμη πιο σημαντικό είναι όταν ένα στοιχείο αναμένει δομημένα δεδομένα. Ίσως χρειαστεί να ρυθμίσετε τον data provider του list control, το οποίο θα είναι μια σειρά από στοιχεία, όχι μόνο μία γραμμή κειμένου. Το ίδιο ισχύει και για τις άλλες ιδιότητες που δέχονται μια σειρά από θέματα, όπως είναι το column properties ενός DataGrid. Το column property σας επιτρέπει να ορίσετε ιδιότητες για μεμονωμένες στήλες, και μπορείτε να δείτε ένα παράδειγμα αυτής της σύροντας ένα DataGrid control για την εφαρμογή σας σε κατάσταση σχεδίασης. Θα δείτε πιθανώς κώδικα όπως αυτό:

```

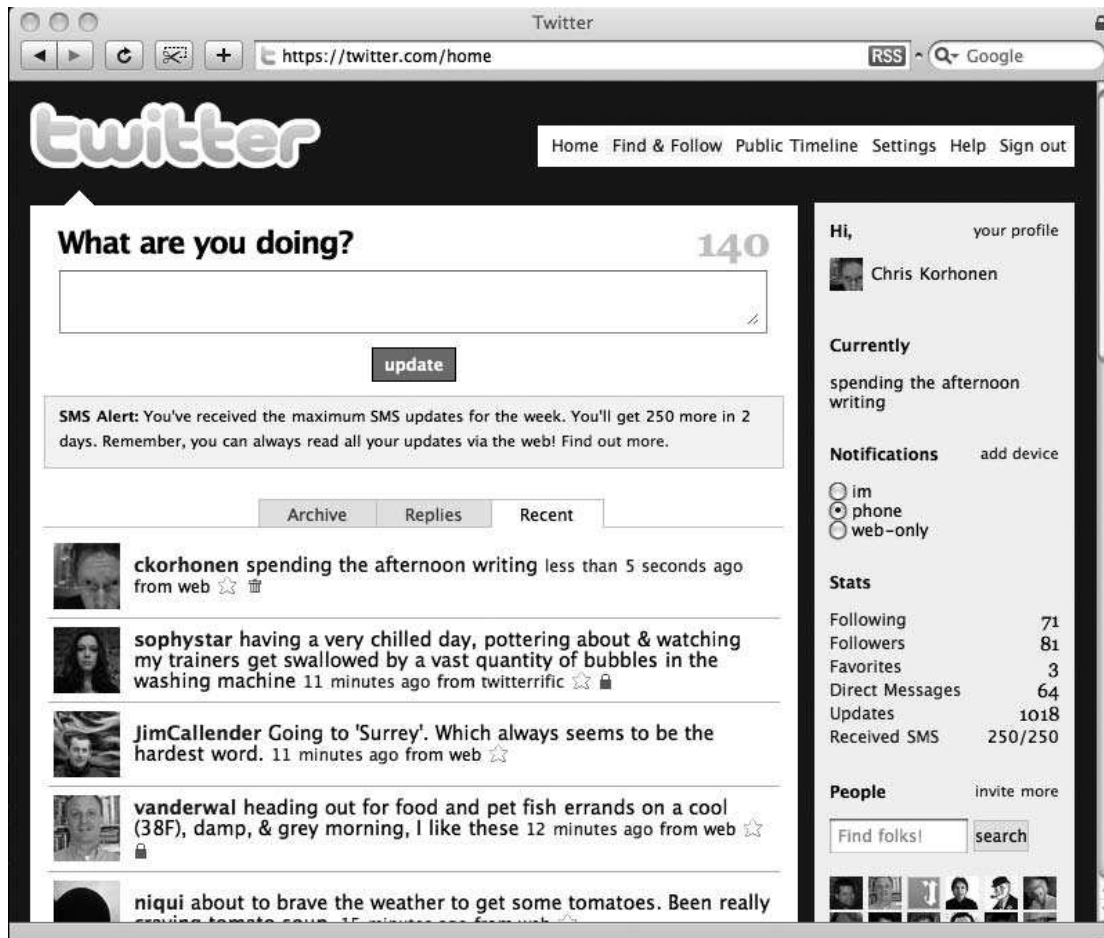
<mx:DataGrid>
  <mx:columns>
    <mx:DataGridColumn headerText="Column 1" dataField="col1"/>
    <mx:DataGridColumn headerText="Column 2" dataField="col2"/>
    <mx:DataGridColumn headerText="Column 3" dataField="col3"/>
  </mx:columns>
</mx:DataGrid>

```

Το column property του DataGrid δεν μπορεί πραγματικά να γραφτεί ως ένα χαρακτηριστικό επειδή αναμένει μια λίστα από DataGridColumn ετικέτες, οι οποίες με τη σειρά τους έχουν τη δική τους ιδιότητες που πρέπει να οριστούν.

3.4 Κατασκευή μιας Flex εφαρμογής

Για παράδειγμα, θα δείξω πώς να κατασκευάσετε ένα απλό front-end για το Twitter

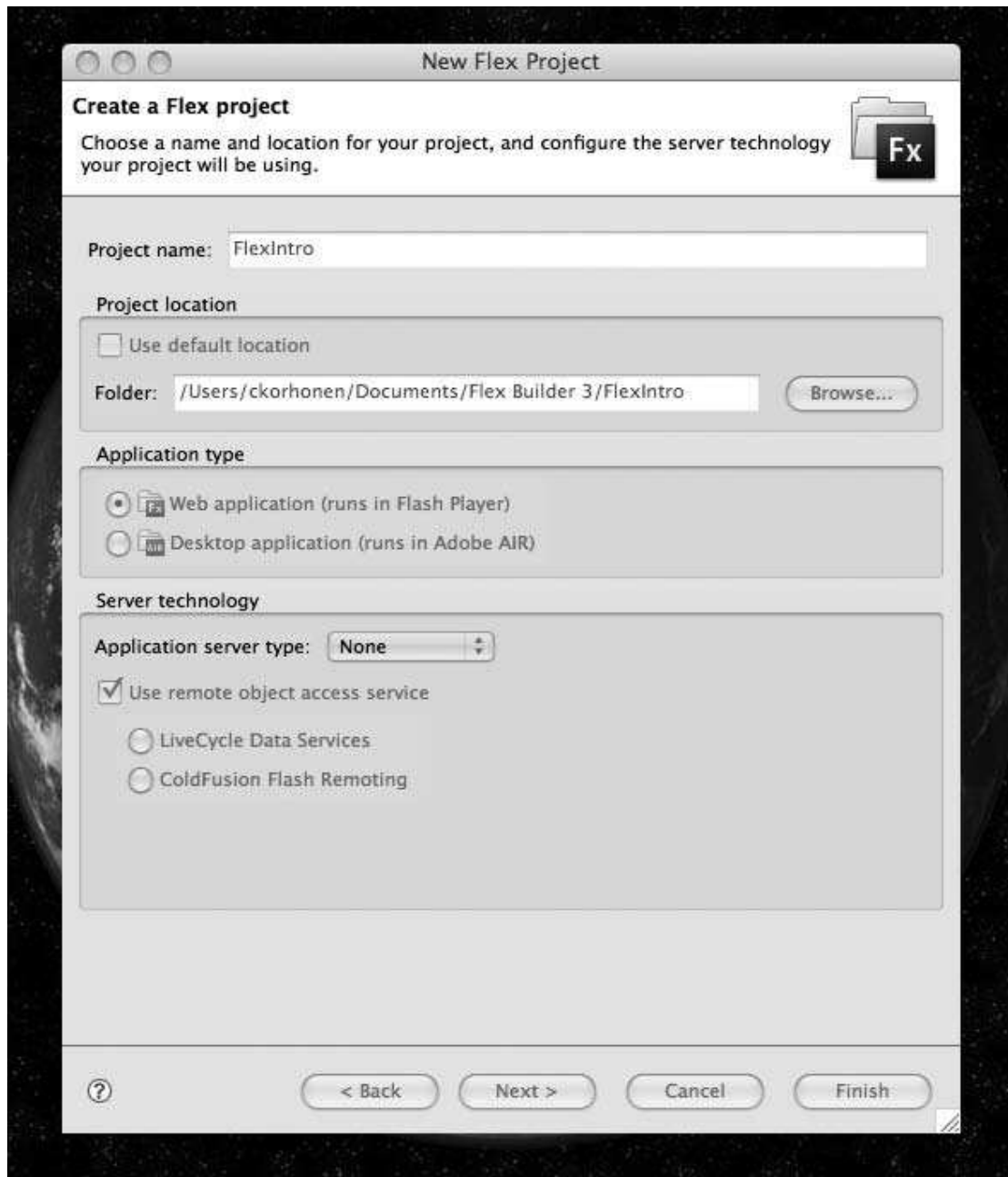


Η αρχική σελίδα του Twitter, δείχνοντας τελευταίες ενημερώσεις από τους φίλους

Για το παράδειγμα που θα κατασκευάσω, θα θέλω να εμφανίσω τη λίστα των ενημερωμένων εκδόσεων από ένα σύνολο φίλων. Τα στοιχεία αυτά παρέχονται ως ένα RSS feed.

3.4.1 Δημιουργία νέου *Flex project*

Το πρώτο βήμα είναι να δημιουργηθεί ένα νέο Flex project, επιλέξτε File → New → Flex Project. Όταν ο Flex Project οδηγός εμφανιστεί θα πρέπει να δώσετε στο Project ένα όνομα και ένα path.



Δημιουργώντας ένα νέο Flex Project

Κάντε κλικ στο κουμπί Finish, και ο Flex Builder θα δημιουργήσει τη δομή φακέλων για την εφαρμογή σας.

Εάν εξετάσετε τη δομή των φακέλων, θα δείτε πολλά αρχεία και φακέλους που έχουν δημιουργηθεί:

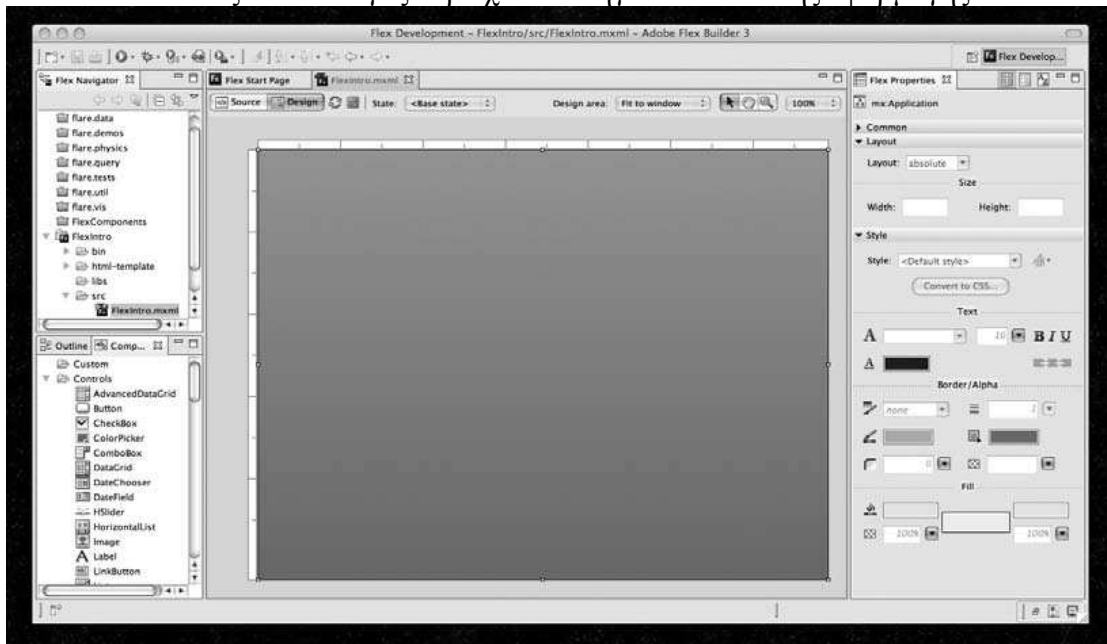
bin: Αυτός είναι ο κατάλογος στον οποίο η compiled version της εφαρμογής σας παράγεται, συμπεριλαμβανομένων του αρχείου SWF και κάθε HTML και JavaScript στοιχείων.

html-πρότυπο: Παραδόξως, αυτό είναι το πρότυπο HTML για την εφαρμογή σας. Μπορείτε να το προσαρμόσετε ανάλογα με τις ανάγκες σας, ή μπορείτε απλά να το αγνοήσετε και να ενσωματωθεί με το SWF αρχείο στις σελίδες σας.

libs: Αυτός ο κατάλογος περιέχει βιβλιοθήκες κώδικα που καθορίζονται κατά

τη δημιουργία του project.

src: Αυτός ο κατάλογος περιέχει τον πηγαίο κώδικα της εφαρμογής.



Το νέο project σας.

3.4.2 Ορίζοντας την εφαρμογή σας

Στον κατάλογο src, θα δείτε ένα αρχείο που ονομάζεται <project name>.mxml. Αυτό είναι ο αρχείο ορισμός της εφαρμογής σας. Αν μεταβείτε από την προβολή σχεδίασης, στην προβολή κώδικα μέσα στον Flex Builder, θα δείτε αυτόν τον κώδικα:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute">
</mx:Application>
```

Στην προβολή σχεδίασης μπορείτε να ξεκινήσετε τον καθορισμό διεπαφής χρήστη της εφαρμογής σας. Από το Component πίνακα, σύρετε ένα στοιχείο Label και ένα στοιχείο List επάνω στον καμβά. Το Label πρόκειται να λειτουργήσει ως τίτλος για την εφαρμογή σας. Μπορείτε να το κάνετε style χρησιμοποιώντας το Property Inspector και στη συνέχεια να το τοποθετήσετε κατάλληλα.

Ο κατάλογος θα γεμίσει με Twitter ενημερώσεις όταν η εφαρμογή ξεκινά. Κάντε την λίστα αρκετά μεγάλη, και ρυθμίστε τα Layout Properties έτσι ώστε το στοιχείο θα είναι πάντα σε μια συγκεκριμένη απόσταση από τις πλευρές της εφαρμογής. Αυτό εξασφαλίζει ότι η διάταξη δεν χαλάει καθώς η εφαρμογή μεγεθύνεται.

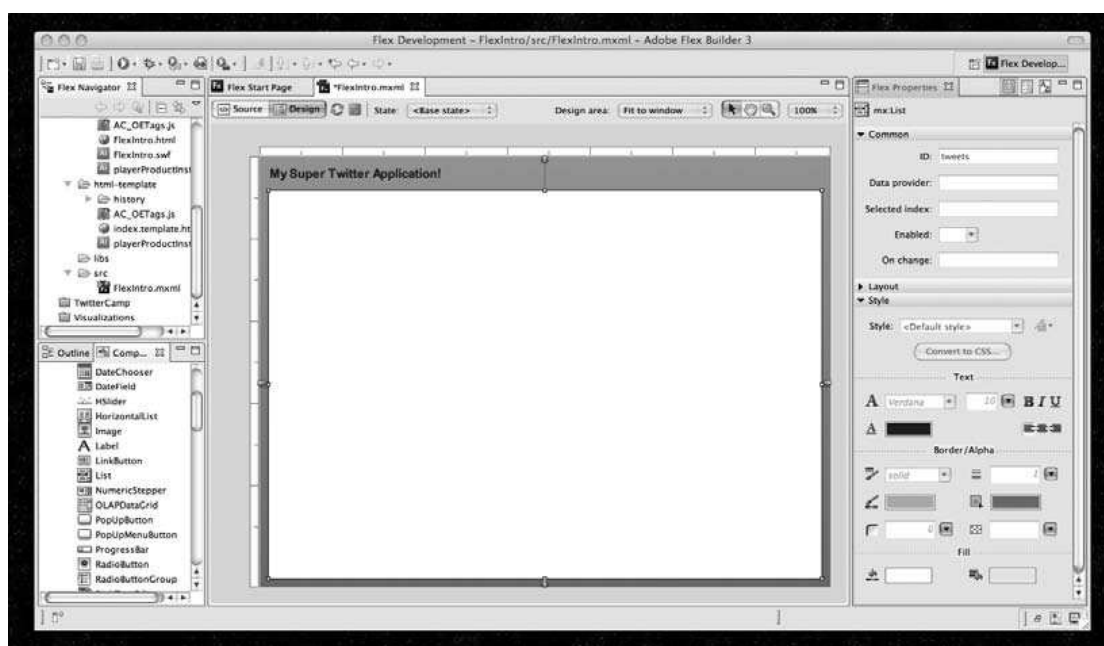
Ένα τελευταίο βήμα πριν τελειώσετε με τις αρχικές διάταξης είναι να δώσετε ένα id attribute στο Property Inspector. Μπορείτε να το κάνετε αυτό απλά με την προσθήκη id = "tweets" στο component's tag.

Εκχωρώντας ένα id attribute σε οποιοδήποτε αντικείμενο ή component σας

επιτρέπει να αναφερθείτε εκεί αργότερα από τον ActionScript κώδικα. Αλλιώς, δεν έχετε κανέναν εύκολο τρόπο συσχέτισης. Σε γενικές γραμμές, ένα αναγνωριστικό θα πρέπει να είναι ένα μοναδικό string, παρόμοιο με ένα όνομα μεταβλητής, και δεν πρέπει να χρησιμοποιείται για τον προσδιορισμό οποιουδήποτε άλλου component μέσα από την εφαρμογή.

Σε Mxml θα πρέπει να έχετε τώρα αυτό το αποτέλεσμα:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
  <mx:Label x="10" y="10" text="My Super Twitter Application!"
    fontFamily="Arial" fontWeight="bold" fontSize="15"/>
  <mx:List left="10" right="10" top="41" bottom="10" id="tweets">
  </mx:List>
</mx:Application>
```



Το αρχικό user interface στο Flex Builder IDE σε Προβολή σχεδίασης

3.4.3 Ολοκλήρωση των δεδομένων σας

Το Twitter, παρέχει ένα RSS feed με τις τελευταίες ενημερώσεις του χρήστη μαζί με τις ενημερώσεις των φίλων. Επιπλέον, το Twitter παρέχει διάφορες API μεθόδους για τις επαφές με τις υπηρεσίες Twitter. Ας δούμε πώς θα φέρετε ένα RSS feed στην εφαρμογή σας.

Το Flex προσφέρει ένα component του HTTPService που επιτρέπει να γίνουν οι HTTP αιτήσεις και τα ανακτηθούν τα δεδομένα απόκρισης. Για να το χρησιμοποιήσετε, μεταβείτε σε προβολή Κώδικα, και καθορίστε το στο MXML αρχείο σας μέσα στο <mx:Application> tag:

```
<mx:HTTPService id="twitterFeed"
  url="https://twitter.com/statuses/friends_timeline/25883.rss" />
```

Όπως μπορείτε να δείτε, καθορίσατε το HTTPService component με τη δική του id χαρακτηριστικό της και επίσης ορίζει το URL που πρέπει να πάρει το RSS feed. Για να μπορέσετε να δείτε όλα τα δεδομένα, θα πρέπει να κάνετε δύο βήματα. Κατ' αρχάς, θα πρέπει να ενεργοποιήσετε το HTTPService component και να δώσετε εντολή που επιχειρεί να φορτώσει το URL του RSS feed. Δεύτερον, πρέπει να πάρετε τα στοιχεία από το response για να γεμίσει η λίστα σας.

Λογικά, θέλετε να προσπαθήσετε να φορτώσετε το RSS feed, όταν η εφαρμογή φορτώνεται αρχικά. Κατά συνέπεια, πρέπει να γίνει τροποποίηση του ορισμού της εφαρμογής για να προσθέσετε ένα event, το οποίο ενεργοποιείται όταν ολοκληρωθεί η φόρτωση του interface:

```
<mx:Application
xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
creationComplete="twitterFeed.send()" >
```

Τώρα που έχετε έναν τρόπο για να φορτώσετε το RSS feed σας, το επόμενο βήμα είναι να αναλύσετε αυτό το έγγραφο XML και να ενημερώσετε το περιεχόμενο του καταλόγου.

Το E4X δίνει την δυνατότητα στον προγραμματιστή να αποκτήσετε πρόσβαση σε ένα έγγραφο XML με ένα πιο φυσικό τρόπο, και παράλληλα προσφέρει καλύτερη απόδοση στο παρασκήνιο. Για παράδειγμα, εδώ είναι μια συμπιεσμένη έκδοση από το RSS feed:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel id="tweets">
    <title>Twitter / ckorhonen with friends</title>
    <link>http://twitter.com/ckorhonen/with_friends</link>
    <description>
      Twitter updates from Chris Korhonen / ckorhonen and
      folks.
    </description>
    <language>en-us</language>
    <ttl>40</ttl>
    <item>
      <title>yezzer: battlestar galactica = tv crack</title>
      <description>
        yezzer: battlestar galactica = tv crack
      </description>
      <pubDate>Sat, 08 Dec 2007 15:29:02 +0000</pubDate>
      <guid>http://twitter.com/yezzer/statuses/481306382</guid>
      <link>http://twitter.com/yezzer/statuses/481306382</link>
    </item>
  </channel>
</rss>
```

Αν θέλετε να αλλάξετε το περιεχόμενο του <title> που συνδέεται με την ενημέρωση του Twitter, θα χρησιμοποιήσετε τη σύνταξη rss.channel.item.title. Είναι πραγματικά τόσο απλό. Στο πλήρες RSS feed, όπου έχετε πολλαπλά nodes <item>, η

πρόσβαση σε δεδομένα γίνεται με παρόμοιο τρόπο. Ωστόσο, αυτή τη φορά έχουμε να κάνουμε με ένα array από items:

```
rss.channel.item[0].title
```

Μπορείτε επίσης να αποκτήσετε πρόσβαση στα attributes συγκεκριμένων nodes χρησιμοποιώντας παρόμοια σύνταξη, αλλά με το @:

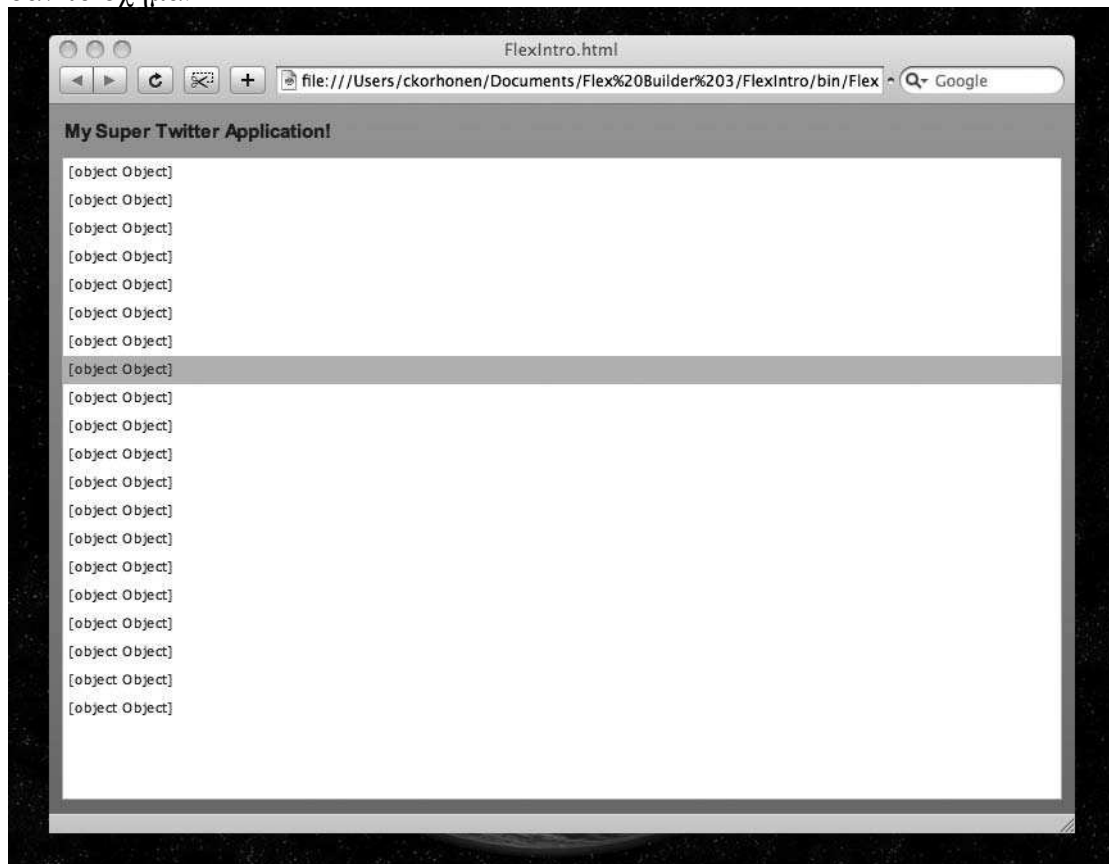
```
rss.channel.@id
```

Για να πάρετε αυτά τα δεδομένα στο List component σας, μπορείτε να χρησιμοποιήσετε κάτι που ονομάζεται data provider. Αυτό είναι ακριβώς το πώς το RSS feed με E4X. Το μόνο που χρειάζεται να κάνετε είναι να πείτε το στοιχείο λίστας ποιος είναι ο παροχέας, χρησιμοποιώντας άγκιστρα ().

```
<mx:List left="10" right="10" top="41" bottom="10" id="tweets"
  dataProvider="{twitterFeed.lastResult.rss.channel.item}" >
</mx:List>
```

Όταν κατεβάσετε το RSS feed, θα παρατηρήσετε πολλά αντικείμενα [object Object] στο component λίστας. Πρόκειται για στοιχεία από το RSS feed σας.

Αν κάνετε compile την εφαρμογή σας και την εκτελέσετε, θα πρέπει να δείτε κάτι σαν το σχήμα:



Η εφαρμογή σας, με τα δυναμικά δεδομένα που φορτώνονται από ένα RSS feed

Κάθε στοιχείο RSS είναι στην πραγματικότητα μια συλλογή από αντικείμενα-τον τίτλο, την περιγραφή, την ημερομηνία, τον οδηγό, και τον σύνδεσμο. Αυτό σημαίνει ότι δεν μπορεί απλώς να αποδίδεται ως ένα στοιχείο λίστας. Μια λύση θα ήταν να χρησιμοποιηθεί ένα στοιχείο DataGrid και όχι μια τυπική συνιστώσα List, η οποία θα βάλει τα αντικείμενα σε διαφορετικές στήλες.

Αντ' αυτού, θα σας δείξουμε πώς να δημιουργήσετε ένα στοιχείο renderer για τη λίστα. Ένα στοιχείο renderer προσφέρει έναν τρόπο καθορισμού των πώς ένα στοιχείο λίστας θα πρέπει να εκδίδεται και είναι συνήθως χτισμένο ως ένα αυτόνομο MXML component. Αυτό σας δίνει μεγάλο έλεγχο, όταν πρόκειται για την εμφάνιση την αισθητική και τη λειτουργικότητα της λίστας σας.

Εάν επιλέξετε File → New, θα δείτε μια επιλογή για να δημιουργήσετε ένα components MXML. Επιλέξτε αυτό, καθώς και το όνομα του component Tweet.

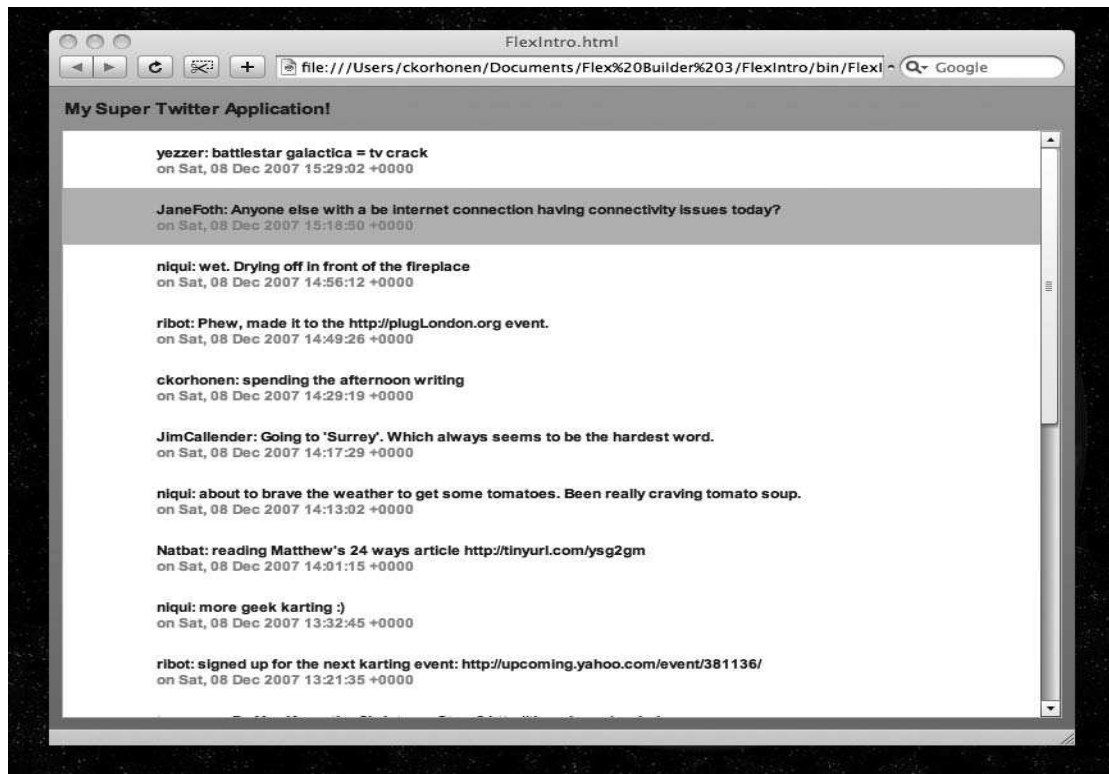
Μπορείτε να δημιουργήσετε ένα MXML component με τον ίδιο τρόπο όπως με την κύρια εφαρμογή σας. Ξεκινήστε με drag μερικών text fields επάνω στον καμβά σε προβολή σχεδίασης. Τακτοποιήστε τα όμως θέλετε. Ας βάλουμε τώρα το κείμενο του κάθε Twitt component και την ημερομηνία που δημοσιεύτηκε.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml"
width="100%" height="50">
  <mx:Text x="69" y="25" text="on {data.pubDate}"
fontFamily="Arial"fontSize="12" fontWeight="bold"
color="#969696"width="620"/>
  <mx:Text y="10" text="{data.title}" width="682"
right="10"fontFamily="Arial"
fontWeight="bold" fontSize="12"/>
</mx:Canvas>
```

Τώρα, αν συνδέσετε το item στο component της λίστα σας και ξανακάνετε compile, θα πρέπει να βλέπε Twitter μηνύματα, ή Tweets, σχήμα 2-6

```
<mx>List left="10" right="10" top="41" bottom="10" id="tweets"
dataProvider="{twitterFeed.lastResult.rss.channel.item}" itemRenderer="Twee
t">
</mx>List>
```

Έτσι, έχετε καταφέρει να πάρετε ένα δυναμικό RSS feed, να φορτώσετε τα δεδομένα του, και να το βάλετε μέσα στην εφαρμογή σας-όλα μέσα σε λιγότερο από περίπου 20 γραμμές κώδικα, το οποίο σας δείχνει ακριβώς πόσο εύκολο είναι να εργαστείτε μέσα στο Flex.



Η εφαρμογή σας, με επιλογή Tweets

3.5 Περνώντας στο Desktop

Συνήθως οι web εφαρμογές έχουν δύο προβλήματα:

Μια web εφαρμογή απαιτεί από το χρήστη να συνδεθεί με το Διαδίκτυο, προκειμένου να έχει πρόσβαση και αξιοποίηση των δεδομένων.

Είναι δύσκολο να υπάρξει συμβατότητα στις desktop εφαρμογές με τους διάφορους web browsers και πλατφόρμες. Παραδοσιακά, έγινε δεκτό ότι δεν μπορείτε να έχετε πρόσβαση σε δεδομένων ενώ δεν ήσασταν συνδεδεμένοι με το Internet. Σήμερα, ωστόσο, η στάση αυτή αλλάζει. Οι χρήστες απαιτούν όλο και περισσότερο οι online εφαρμογές να αποθηκεύουν τα δεδομένα τους.

Για παράδειγμα, ένας χρήστης μπορεί να χρησιμοποιήσει το Google Mail για τα e-mails του, το Flickr για αποθήκευση φωτογραφιών, και μια εφαρμογή ημερολογίου όπως το 30 Boxes για να παραμείνει οργανωμένος. Επιπλέον, μπορούν να αποθηκεύουν τις επαφές στο Google Mail παράλληλα με τις instant-messaging συνομιλίες. Χωρίς σύνδεση, ο χρήστης δεν έχει πρόσβαση σε οποιαδήποτε από αυτές τις λειτουργίες ή τα δεδομένα. Για κάποιους, αυτό ενδέχεται να μην είναι μεγάλο πρόβλημα, αλλά για άλλους θα μπορούσε να είναι ένα σημαντικό μειονέκτημα.

Task	Desktop Application	Web-Based Alternative
E-mail	Outlook, Mac Mail, Thunderbird	Google Mail, Yahoo Mail
Calendar	Outlook, iCal, Lotus Notes	Google Calendar, 30 Box
Word processing	Microsoft Office, Open Office	Google Documents
Spreadsheets	Microsoft Office, Open Office	Google Spreadsheets
Photos	iPhoto, Picasa	Flickr
Project management/ organization	OmniProject, Microsoft Project	Basecamp

Ο πίνακας αναγράφει τις ποιο συνήθεις web-based εναλλακτικές για τις εφαρμογές web.

Φυσικά, τα εργαλεία που μπορούν να χρησιμοποιηθούν για την πρόσβαση δεδομένων χωρίς σύνδεση σε ορισμένες από αυτές υπάρχουν. Για παράδειγμα, το Google

Mail προσφέρει πρόσβαση μέσω POP, έτσι ώστε οι χρήστες μπορούν να κατεβάσουν τα μηνύματα του ηλεκτρονικού ταχυδρομείου τους σε ένα desktop client. Λογισμικό επίσης, υπάρχει για να συγχρονίσετε το ημερολόγιο Google με τα ποιο κοινά ημερολόγια γραφείου.

Το μειονέκτημα αυτών είναι ότι είναι αναγκασμένος ο χρήστης να διατηρεί τα δεδομένα του μέσω πολλαπλών διαφορετικών λογισμικών. Έτσι, ένας χρήστης μπορεί να έχει πρόσβαση στα δεδομένων χωρίς σύνδεση, αλλά δεν μπορεί να έχει πρόσβαση σε οποιοδήποτε λειτουργία μιας web εφαρμογής. Το Google Mail είναι ένα καλό παράδειγμα-κατά τη λήψη των Mails σας χρησιμοποιώντας POP, θα χάσετε όλη τη λειτουργικότητα labeling και filtering που προσφέρεται από την web εφαρμογή, και επίσης δεν θα έχετε πλέον πρόσβαση στο βιβλίο διευθύνσεων σας ή στην πολύ σημαντική λειτουργία search.

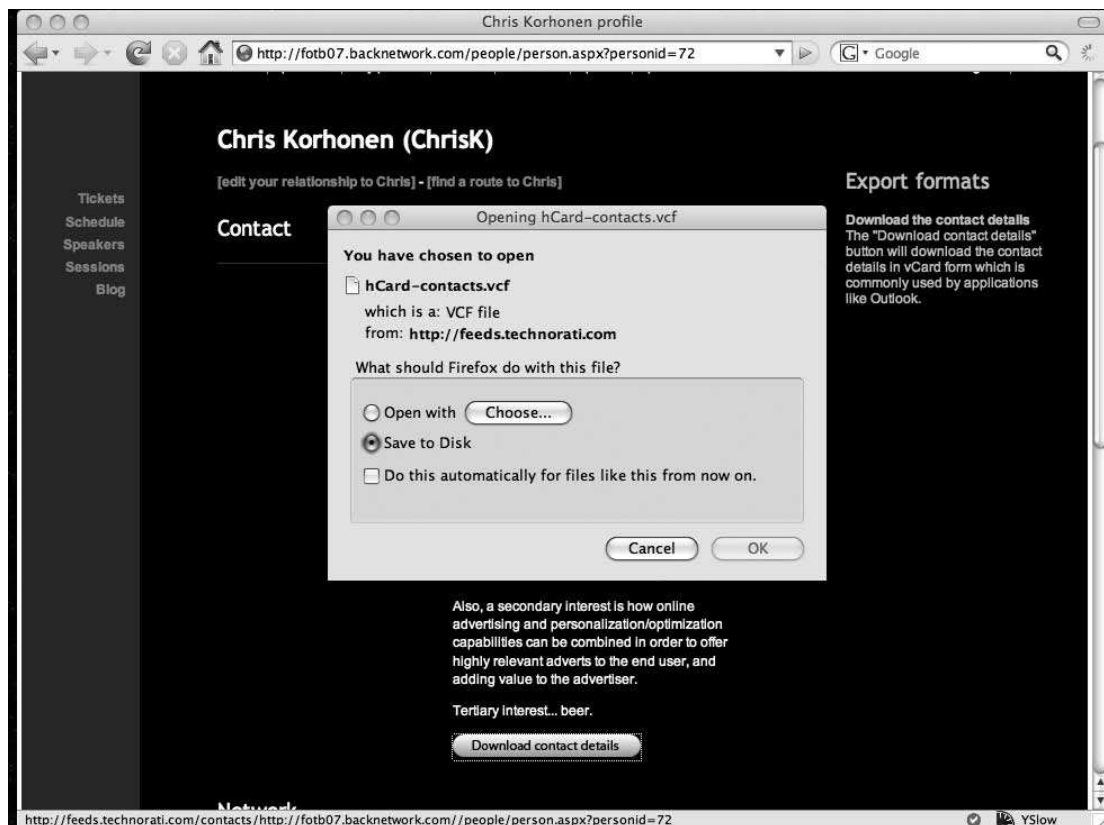
Η μόνη λύση, είναι να δημιουργήσουμε μια desktop εφαρμογή που προσφέρει λειτουργικότητα είτε ο χρήστης είναι online ή offline.

Φανταστείτε να επιθυμούσε ένας χρήστης να έχει τη δυνατότητα να εισάγει ορισμένα στοιχεία από την web εφαρμογή στο βιβλίο διευθύνσεων (Σχήμα 11-1). Δεν υπάρχει αξιόπιστη μέθοδος για αυτό, η καλύτερη μέθοδος είναι να διατεθούν τα δεδομένα, ως vCard, ένα τυποποιημένο έντυπο για τις καταχωρήσεις του βιβλίου διευθύνσεων. Πέρα αυτό, δεν είναι τίποτα σίγουρο.



Ένα παράδειγμα σε Mac OS ηλεκτρονικού ταχυδρομείου όπου δεδομένα εντοπίζονται και εισάγονται στην βιβλίο διευθύνσεων του συστήματος

Πώς ο browser του χρήστη θα χειριστεί μια vCard; Θα προσπαθήσει να την δώσει σε μια εξωτερική εφαρμογή; Πώς το λειτουργικό σύστημα του χρήστη θα χειριστεί την vCard; Έχει ο χρήστης το κατάλληλο file associations set up για να χειριστεί την vCard; Έχει ο χρήστης συμβατή εγκατεστημένη εφαρμογή; Με όλα αυτά τα πιθανά σημεία αποτυχίας, μπορεί απλώς να είναι πιο εύκολο να ζητήσει από το χρήστη να αντιγράψει και να επικολλήσει στο βιβλίο διευθύνσεων τα δεδομένα με το χέρι-αν και αυτό δεν θα ήταν η καλύτερη επιλογή.



Μια προσπάθεια να χρησιμοποιηθεί μια vCard στο Firefox, το πρόγραμμα δεν γνωρίζει τι να κάνει!

Εκτός από την ανταλλαγή δεδομένων μεταξύ των εφαρμογών, μια διαδικτυακή εφαρμογή δεν μπορούμε να ελπίζουμε ότι θα συνεργαστεί με το τοπικό σύστημα αρχείων ή το λειτουργικό σύστημα. Για να επιτευχθεί αυτό, θα πρέπει να έχετε μια εφαρμογή desktop.

3.5.1 Γιατί να αναπτύξω εφαρμογή desktop;

Από τεχνική σκοπιά, το desktop μπορεί να προσφέρει λύσεις στα προβλήματα της ενσωμάτωσης με άλλες εφαρμογές και με το λειτουργικό σύστημα του χρήστη. Μια desktop εφαρμογή μπορεί επίσης να προσφέρει δυνατότητες που να παρέχουν πρόσβαση δεδομένων χωρίς σύνδεση σε μια διαδικτυακή εφαρμογή. Αλλά γιατί θα πρέπει να εστιαστείτε στην ανάπτυξη desktop εφαρμογών;

Η απλή απάντηση είναι ότι υπάρχουν πολλά πράγματα που μπορεί να προσφέρει μια desktop εφαρμογή, πράγματα που είτε δεν μπορούν να εφαρμοστούν είτε δεν μπορούν να υλοποιηθούν σε μια διαδικτυακή εφαρμογή.

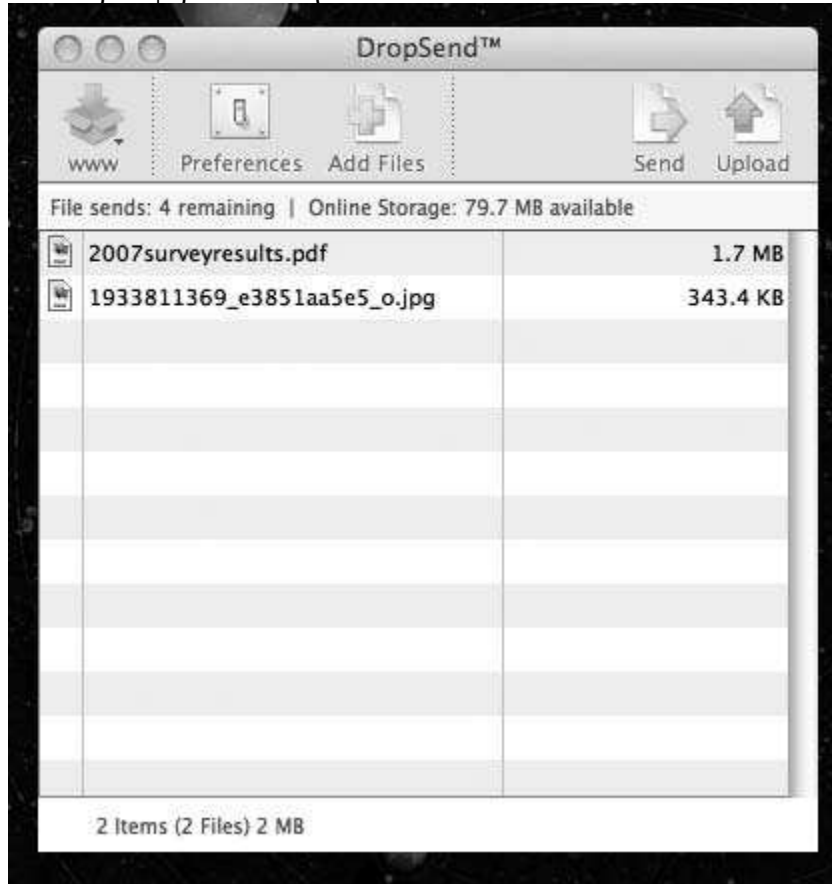
Για παράδειγμα, ρίξτε μια ματιά στο DropSend (www.dropsend.com), μια διαδικτυακή εφαρμογή που επιτρέπει στους χρήστες να ανεβάσουν μεγάλα αρχεία και να τα στείλουν με e-mail. Ο παραλήπτης θα λάβει ένα σύνδεσμο όπου μπορεί να κατεβάσει το αρχείο.

Ένα από τα κύρια καθήκοντα της εφαρμογής είναι να επιτρέψει στους χρήστες να ανεβάζουν αρχεία, μια λειτουργία που γίνεται παραδοσιακά με άσχημο τρόπο μέσα στις web εφαρμογές. Λόγω της περιορισμένης λειτουργίας του web browser, ένας

χρήστης πρέπει να επιλέξετε κάθε αρχείο ξεχωριστά για να το φορτώσει. Κατά την αποστολή μεγάλου αριθμού αρχείων, αυτή η διαδικασία γίνεται δύσκολη. Ως εκ τούτου, οι δημιουργοί του DropSend αποφάσισαν να δημιουργήσουν μια desktop εφαρμογή.

Αυτή η desktop έκδοση του DropSend επιτρέπει στο χρήστη να κάνει απλά drag and drop τα αρχεία επάνω στην εφαρμογή κάνοντας τα έτσι έτοιμα για αποστολή.

Αυτό προσφέρει απλότητα και ευκολία



Ο desktop client DropSend. Ένας χρήστης απλά ρίχνει τα αρχεία πάνω από το παράθυρο και τα αποστέλλει για online αποθήκευση.

3.6 Δημιουργία AIR εφαρμογής

Το AIR είναι συμβατό με όλα τα λειτουργικά συστήματα, δίνοντας την δυνατότητα στον προγραμματιστή να τον απασχολεί μόνο το interface και η λειτουργικότητα της εφαρμογής και όχι το πώς θα διαβάσει ένα αρχείο από ένα σύστημα αρχείων Mac OS ή τον τρόπο εμφάνισης ενός notification στα Windows Vista.

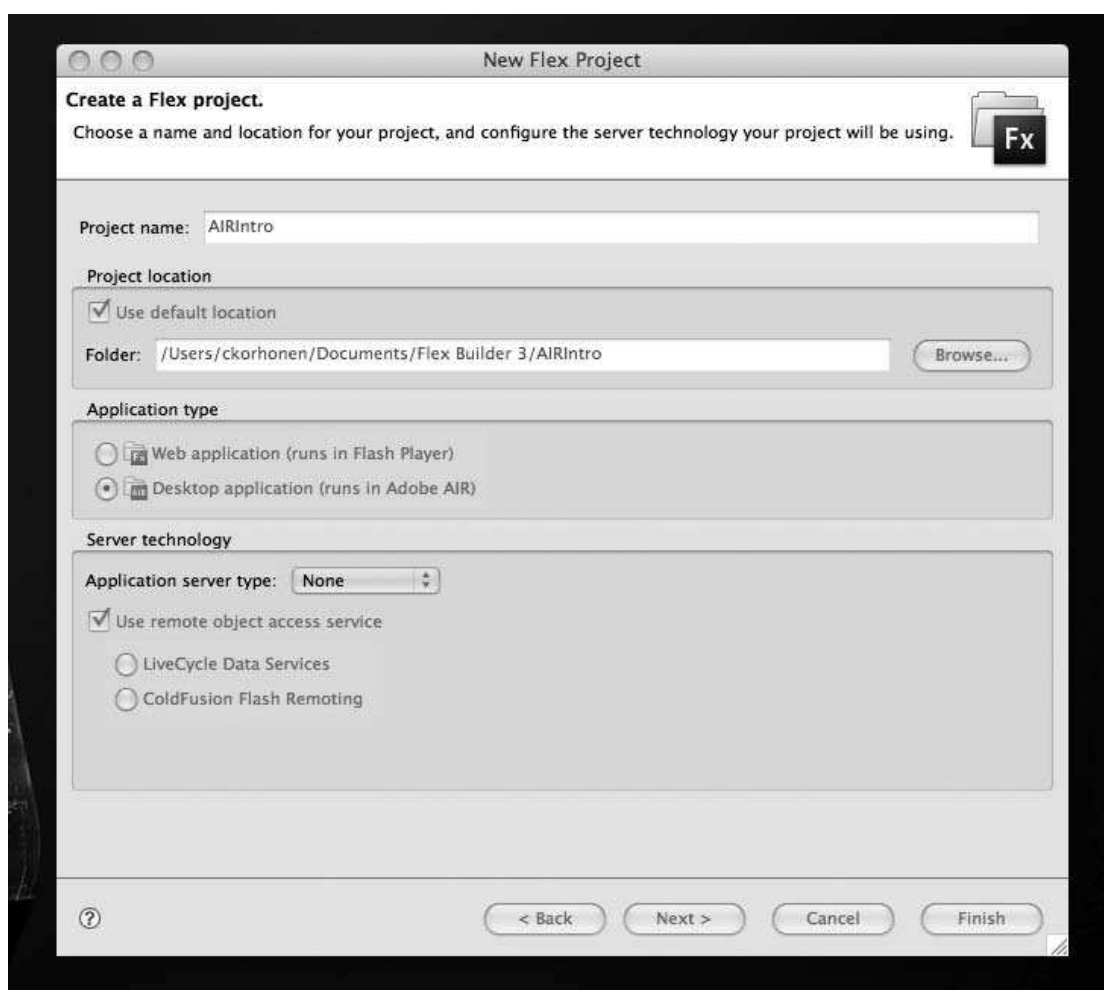
Το AIR επίσης, προσφέρει τη λειτουργικότητα που οι προγραμματιστές μπορούν να αξιοποιήσουν σε Flex ή HTML / JavaScript εφαρμογές, οι οποίες είναι συχνά επιθυμητή.

Αυτό περιλαμβάνει την πρόσβαση στον τοπικό σύστημα αρχείων και την αποθήκευση και την ικανότητα να ανταποκρίνεται σε drag-and-drop αλληλεπιδράσεις.

Για να δημιουργήσουμε μια Flex-based εφαρμογή AIR, χρειάζεστε ένα αντίγραφο του Flex Builder 3, ή μπορείτε να χρησιμοποιήσετε τη δωρεάν AIR SDK (www.adobe.com/go/getairsdk), η οποία παρέχεται από την εταιρεία Adobe και μπορεί να χρησιμοποιηθεί σε συνδυασμό με κάθε IDE, προκειμένου να μπορεί να κάνει compile και debug την AIR εφαρμογή σας. Χρειάζεστε επίσης ένα αντίγραφο του AIR εγκατεστημένο καθώς σχεδιάζετε την AIR εφαρμογή σας.

3.6.1 Ξεκινώντας νέο project

Στον Flex Builder, θα ακολουθήσετε μια παρόμοια διαδικασία που χρησιμοποιήσατε για να δημιουργήσετε μια νέα Flex εφαρμογή. Πρώτα επιλέξτε File → Flex Project και στη συνέχεια επιλέξτε Desktop Application ως τύπος εφαρμογής στον οδηγό που εμφανίζεται



Δημιουργώντας μια νέα εφαρμογή AIR

Όπως και πριν, μπορείτε να αφήσετε τις υπόλοιπες επιλογές του οδηγού όπως είναι εκτός αν θέλετε να αλλάξετε file locations, να κάνετε import βιβλιοθήκες, ή να ενσωματώσετε την εφαρμογή σας απευθείας σε έναν local server.

Όταν κάνετε κλικ στο κουμπί Finish, ο Flex Builder θα δημιουργήσει τη δομή εφαρμογής. Η δομή των φακέλων που δημιουργούνται στο AIR για την εφαρμογή σας είναι σχεδόν ταυτόσημη με αυτό που θα παίρνατε αν κατασκευάζατε μια Flex

εφαρμογή.

Έχετε το src directory σας, όπου περιέχει τον πηγαίο κώδικα, το libs directory σας για οποιαδήποτε εξωτερική βιβλιοθήκη μπορεί να θέλετε να χρησιμοποιήσετε και τον κατάλογο bin σας, όπου η εφαρμογή σας θα δημιουργηθεί.

Μπορείτε να παρατηρήσετε ότι δεν έχετε πλέον έναν html-template κατάλογο, εφόσον τα αρχεία που χρησιμοποιούνται ως βάση HTML για τη σελίδα σας θα μένουν κανονικά. Ο λόγος είναι πολύ απλός: δεν κατασκευάζετε πλέον την εφαρμογή σας σε μια ιστοσελίδα. Αντ' αυτού, θα τρέχει στην επιφάνεια εργασίας.

Αν ρίξετε μια ματιά στον src κατάλογο, θα διαπιστώσετε ότι έχετε δύο αρχεία, το οικείο αρχείο σας MXML και ένα νέο αρχείο που λήγει σε -app.xml, το οποίο είναι ο descriptor της εφαρμογής AIR που περιέχει τις ειδικές παραμέτρους που χρησιμοποιούνται κατά την εγκατάσταση και τη λειτουργία εφαρμογών. Ας ανοίξουμε το MXML αρχείο:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute">
</mx:WindowedApplication>
```

Αυτό πρέπει να μοιάζει με μια Flex εφαρμογή, με τη μόνη διαφορά ότι το πρωταρχικό όνομα ετικέτας. Σε Flex, αυτό θα ήταν <mx:Application>. Με το AIR, είναι <mx:WindowedApplication>. Η διαφορά είναι ότι WindowedApplication κληρονομεί πρόσθετη λειτουργικότητα που εφαρμόζεται ιδιαίτερα στις εφαρμογές desktop.

Επειδή δεν υπάρχουν θεμελιώδεις διαφορές μεταξύ Flex εφαρμογών που κατασκευάζονται για το Web και εφαρμογές για την επιφάνεια εργασίας, η μετατροπή εφαρμογών που ήδη υπάρχουν είναι μια σχετικά ανώδυνη διαδικασία. Εάν σας παραπέμψω στο Twitter που αναπτύχθηκε στο κεφάλαιο 2 και αντιγράψετε τα αρχεία στο νέο Flex Project σας, μπορείτε να γρήγορα και απλά να το μετατρέψετε σε AIR.

Εάν ανοίξετε το αρχείο MXML για το Flex της εφαρμογής σας, θα δείτε τα εξής:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
    creationComplete="twitterFeed.send()">
    <mx:HTTPService id="twitterFeed"
        url="http://www.twitter.com/statuses/friends_timeline/25883.rss" />

    <mx>List left="10" right="10" top="41" bottom="62" id="tweets"
        dataProvider="{twitterFeed.lastResult.rss.channel.item}"
        itemRenderer="Tweet">
    </mx>List>

    <mx:Label x="10" y="10" text="My Super Twitter Application!"
        fontFamily="Arial" fontWeight="bold"
        fontSize="15"/>
    <mx:TextArea bottom="10" left="10" right="112" id="tweet_txt"
        maxChars="160"/>
```



```

<mx:Button label="Post" bottom="10" right="10" height="44"
    width="94"
    id="submit_btn" click="postData()"/>

<mx:Script>
    <![CDATA[
        import mx.rpc.events.ResultEvent;
        import com.dynamicflash.util.Base64;

        public function postData():void{
            var request:HTTPService = new HTTPService();
            request.url = "http://twitter.com/statuses/
                update.xml";
            request.method = "POST";
            request.addEventListener(ResultEvent.RESULT,
                handleComplete);

            var username:String = "username";
            var password:String = "password";
            var encodedCredentials:String = Base64.encode(
                username + ":" + password);

            var headerArray =
                new Array(
                    new URLRequestHeader("Authentication",
                        "Basic " + encodedCredentials));

            request.headers = headerArray;
            request.request = { status: tweet_txt.text };
            request.send();
        }

        public function handleComplete(e:Event):void{
            tweet_txt.text = "";
            twitterFeed.send();
        }
    ]]>
</mx:Script>
</mx:Application>

```

Επικολλήστε αυτόν τον κώδικα σε αρχείο MXML για AIR εφαρμογή σας, αλλάζοντας <mx:Application> με <mx:WindowedApplication>:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
    creationComplete="twitterFeed.send()">
    <mx:HTTPService id="twitterFeed"
        url="http://www.twitter.com/statuses/friends_timeline/25883.rss" />
    <mx>List left="10" right="10" top="41" bottom="62" id="tweets"
        dataProvider="{twitterFeed.lastResult.rss.channel.item}"

```

```

        itemRenderer="Tweet">
</mx:List>

<mx:Label x="10" y="10" text="My Super Twitter Application!"
    fontFamily="Arial" fontWeight="bold"
    fontSize="15"/>

<mx:TextArea bottom="10" left="10" right="112" id="tweet_txt"
    maxChars="160"/>
<mx:Button label="Post" bottom="10" right="10" height="44"
    width="94"
    id="submit_btn" click="postData()"/>

<mx:Script>
    <![CDATA[
        import mx.rpc.events.ResultEvent;
        import com.dynamicflash.util.Base64;

        public function postData():void{
            var request:HTTPService = new HTTPService();
            request.url = "http://twitter.com/statuses/
                update.xml";
            request.method = "POST";
            request.addEventListener(ResultEvent.RESULT,
                handleComplete);

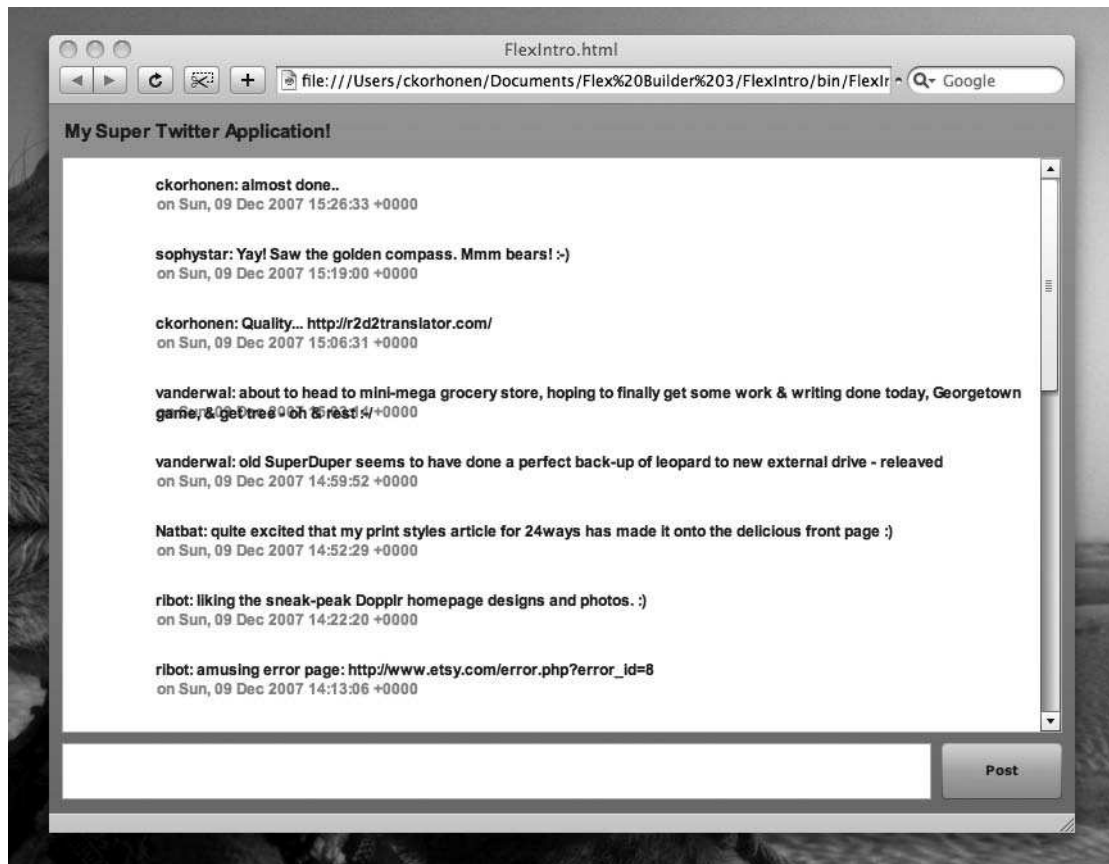
            var username:String = "username";
            var password:String = "password";
            var encodedCredentials:String =
                Base64.encode(username + ":" + password);

            var headerArray =
                new Array(new
                URLRequestHeader("Authentication",
                    "Basic " + encodedCredentials));

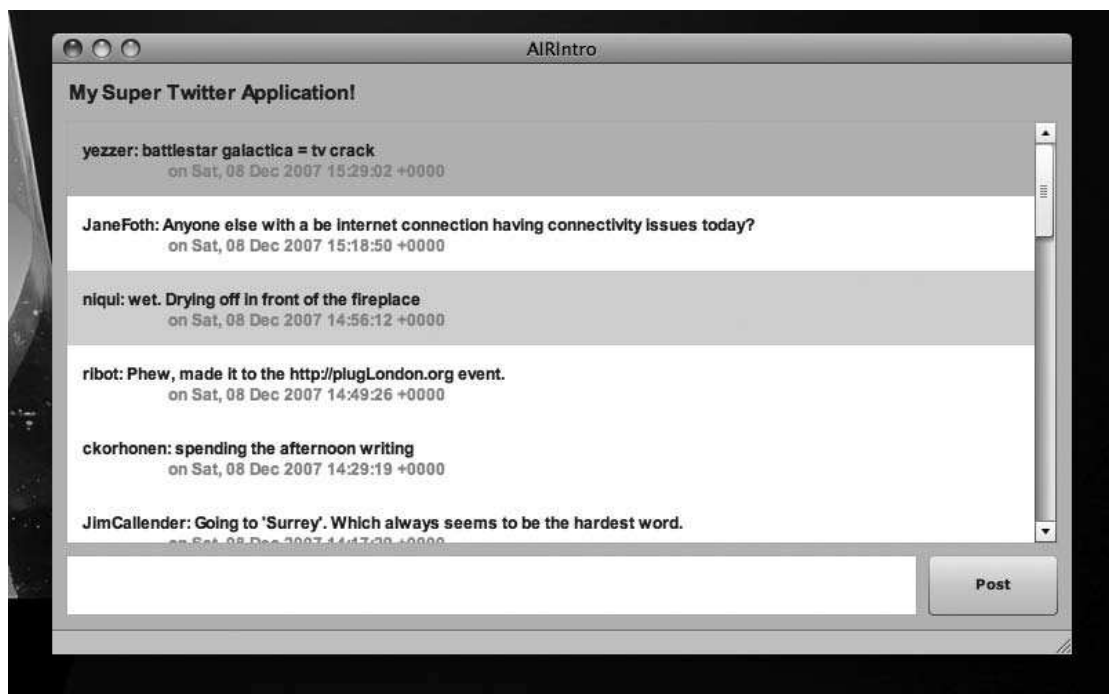
            request.headers = headerArray;
            request.request = { status: tweet_txt.text };
            request.send();
        }
        public function handleComplete(e:Event):void{
            tweet_txt.text = "";
            twitterFeed.send();
        }
    ]]>
</mx:Script>
</mx:WindowedApplication>

```

Τώρα μπορείτε να κάνετε compile και να τρέξετε την εφαρμογή, και θα δείτε ότι λειτουργεί σε desktop. Συγκρίνετε την έκδοση του web προς την έκδοση desktop.



Η Twitter εφαρμογή σας που τρέχει σε ένα web browser



Η Twitter εφαρμογή σας τρέχει στην επιφάνεια εργασίας

Είναι πραγματικά απλό να μετατρέψετε υπάρχουσες εφαρμογές σε εφαρμογές desktop, και αυτό ήταν ένα από τα βασικά πλεονεκτήματα της Adobe κατά τον σχεδιασμό και την κατασκευή AIR εφαρμογών. Έτσι οι προγραμματιστές web

μπορούν να χρησιμοποιήσουν τις ικανότητες τους χωρίς να απαιτείται μαζική επανεγγραφή κώδικα. Παρόλο που μπορεί να είναι καλύτερο να περνούν περισσότερο χρόνο προσαρμόζοντας εφαρμογές για το desktop, υπάρχει η δυνατότητα να μπορούν να εισάγουν γρήγορα υλικό από το Web.

3.6.2 Καθορισμός και προσαρμογή εφαρμογής

Τώρα που προγραμματίζετε για την επιφάνεια εργασίας, έχετε περισσότερο έλεγχο πάνω στην εφαρμογή σας. Αυτό περιλαμβάνει συγκεκριμένα στοιχεία που εμφανίζονται στο χρήστη. Το μέγεθος, τη θέση και εμφάνιση του παραθύρου της εφαρμογής και ακόμη και πώς η εφαρμογή ενσωματώνεται με το τοπικό σύστημα αρχείων όσον αφορά το χειρισμό διαφορετικών τύπων αρχείων.

Όλα αυτά τα στοιχεία καθορίζονται στο app.xml αρχείο σας.

Αν κοιτάξετε το αρχείο που δημιουργήθηκε με την Twitter εφαρμογή, μπορείτε να δείτε:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.0.M6">
  <id>AIRIntro</id>
  <filename>AIRIntro</filename>
  <version>v1</version>

  <initialWindow>
    <content>
      [This value will be overwritten by Flex Builder in
      the output app.xml]
    </content>
  </initialWindow>
</application>
```

Αν κοιτάξετε μέσα στο εγγράφου XML, το πρώτο πράγμα που θα δείτε είναι το XML namespace attribute (xmlns), το οποίο χρησιμοποιείται για να υποδηλώσει τη συγκεκριμένη έκδοση του AIR που απαιτείται για να τρέξει η εφαρμογή. Εδώ χρειαζόμαστε AIR 1.0 Milestone 6 εγκατεστημένη έκδοση ή μεγαλύτερη, διότι αυτή είναι η έκδοση του AIR SDK που χρησιμοποιούμε εδώ.

Εντός του <application>, έχετε επιπλέον κόμβους που περιγράφουν διαφορετικές ιδιότητες της εφαρμογής:

Το <id> χρησιμοποιείται για να προσδιορίσει ακριβώς την εφαρμογή σας στο AIR. Συνήθως είναι το όνομα της εφαρμογής σας.

Το όνομα αρχείου που προκύπτει από την εφαρμογή AIR ορίζεται από το <filename>. Αυτό μπορεί να είναι ό, τι θέλετε, εφόσον δεν περιέχει ειδικούς χαρακτήρες που το λειτουργικό σύστημα δεν μπορεί να επιτρέψει να χρησιμοποιηθούν σε ονόματα αρχείων. Το όνομα του αρχείου λειτουργεί επίσης ως τίτλο το προεπιλεγμένο παράθυρο για την εφαρμογή σας εάν δεν παρέχεται αλλού.

Η <version> δηλώνει την έκδοση της εφαρμογής σας. Μπορεί να πάρει οποιαδήποτε μορφή θέλετε.

Τέλος, το <initialWindow>, το οποίο περιγράφει το αρχικό παράθυρο που θα

προβληθεί όταν η αίτηση ξεκινήσει. Στην περίπτωση αυτή, το περιεχόμενο για αυτό το παράθυρο αυτόματα προστίθεται από το Flex Builder κατά το compile ωστόσο, αυτό μπορεί να αντικατασταθεί από ένα SWF ή HTML αρχείο, αν το επιθυμείτε.

3.6.2.1 Τροποποιώντας το παράθυρο εφαρμογής

Αν σκεφτεί κανείς το αρχικό παράθυρο, μπορεί να αποφασίσει αν θέλετε να τροποποιήσει τον τρόπο που το παράθυρο εμφανίζεται και συμπεριφέρεται. Ευτυχώς, υπάρχουν πολλές προαιρετικές παράμετροι που μπορείτε να χρησιμοποιήσετε για να το κάνετε αυτό. Το πρώτο πράγμα που μπορείτε να κάνετε είναι να αλλάξετε τον τίτλο του παραθύρου της εφαρμογής σας. Στην περίπτωση αυτή, στην Twitter εφαρμογή μας:

```
<title>My Twitter Application</title>
```

Τα AIR παράθυρα εφαρμογών χρησιμοποιούν την προεπιλεγμένη εμφάνιση (που συχνά αναφέρεται ως Chrome system), από το λειτουργικό σύστημα, ώστε στα Windows τα Maximize / Minimize / Close κουμπιά θα είναι στα δεξιά, και σε Mac OS X θα είναι στα αριστερά. Από προεπιλογή στο systemChrome, έχουμε:

```
<systemChrome>standard</systemChrome>
```

Για να προσαρμόσετε μια εφαρμογή, όπως να αφαιρέσετε την γραμμή τίτλου του παραθύρου τελείως ή αντικαταστήσετε ένα skin, μπορείτε να ορίσετε την τιμή του systemChrome property σε none.

Αυτό είναι ιδιαίτερα σημαντικό αν κατασκευάζετε ένα widget πάνω στην πλατφόρμα AIR που συχνά επιδιώκετε να ταιριάζουν οπτικά με το λειτουργικό σύστημα.

Με τον καθορισμό της transparent ιδιότητας, μπορείτε να ελέγξετε αν παράθυρο της εφαρμογής σας μπορεί να έχει διαφανή τμήματα. Δυστυχώς, αυτή είναι διαθέσιμη μόνο όταν δεν χρησιμοποιείτε το standard system Chrome.

```
<transparent>>false</transparent>
```

Με αυτή την επιλογή ενεργοποιημένη, μπορείτε να ελέγξετε τη διαφάνεια, ορίζοντας την ιδιότητα alpha του WindowedApplication.

Εάν θέλετε να προσαρμόσετε τη διαφάνεια μόνο των στοιχείων που περιλαμβάνονται στην εφαρμογή και όχι όλο το παράθυρο της εφαρμογής, τότε δεν χρειάζεται να ρυθμίσετε αυτήν την ιδιότητα.

Αν θέλουμε να αναπτύξουμε μια εφαρμογή που εκτελεί ενδεχομένως εργασίες στο παρασκήνιο, ή από προεπιλογή να ελαχιστοποιείται στο system tray, τότε μπορείτε να χρησιμοποιήσετε την visible ιδιότητα που καθορίζει αν κάποιο παράθυρο εμφανίζεται όταν ξεκινάει η εφαρμογή:

```
<visible>>true</visible>
```

Η λειτουργικότητα ενός παραθύρου μπορεί να ελεγχθεί, δίνοντας τον έλεγχο στον προγραμματιστή για το αν ένας χρήστης μπορεί να κάνει ελαχιστοποίηση, μεγιστοποίηση, ή να αλλάξει το μέγεθός του. Επιπλέον, μπορείτε επίσης να επιλέξετε το ελάχιστο και το μέγιστο μέγεθος της εφαρμογής, προκειμένου να αποφευχθεί να χαλάσει η διάταξη της διεπαφή και να είναι ακατάλληλο προς χρήση σε ακραίες περιπτώσεις. Στην περίπτωση εφαρμογών με σταθερή διάταξη, ο περιορισμός της μεταβολής του μεγέθους της εφαρμογής μπορεί επίσης να είναι επιθυμητός.

```
<minimizable>>false</minimizable>  
<maximizable>>false</maximizable>  
<resizable>>true</resizable>  
<minSize>100,100</minSize>  
<maxSize>600,800</maxSize>
```

Τέλος, μπορείτε να ρυθμίσετε επίσης οι αρχικές ιδιότητες του παραθύρου σας, έτσι ώστε να εμφανίζεται σε μια ορισμένη θέση και σε ένα ορισμένο μέγεθος:

```
<width>400</width>  
<height>500</height>  
<x>20</x>  
<y>20</y>
```

Από προεπιλογή, τα παράθυρα εφαρμογής θα πρέπει να εμφανίζονται στο κέντρο της οθόνης, παίρνοντας το πλάτος και το ύψος της Flex εφαρμογής.

3.6.2.2 Εργασία με διαφορετικούς τύπους περιεχομένου

Για τη δημιουργία AIR εφαρμογών, ο προγραμματιστής σε καμία περίπτωση δεν περιορίζεται στη χρήση Flex. Υπάρχουσες εφαρμογές Flash μπορούν να μετατραπούν σε AIR εφαρμογές. Για να επιτευχθεί αυτό, απλώς αλλάζει το `<Content>` μέσα στο `initialWindow` ώστε να δείξει σε ένα τοπικό αρχείο SWF, κάπως έτσι:

```
<content>myswf.swf</content>
```

Όταν κάνετε `compile` την εφαρμογή σας, θα πάρει το αρχείο SWF σαν `content` για το αρχικό παράθυρο.

Ο μόνος περιορισμός όταν μετατρέπετε Flash περιεχόμενο σε μια desktop εφαρμογή είναι ότι αν έχετε μια ActionScript 2,0-based εφαρμογή και δεν θα έχετε τη δυνατότητα πρόσβασης σε οποιοδήποτε από τα AIR APIs επειδή εφαρμόζονται σε ActionScript 3.0. Αυτό σημαίνει ότι οι εν λόγω εφαρμογές δεν θα είναι σε θέση να έχουν πρόσβαση στο τοπικό σύστημα αρχείων, τη χρήση drag-and-drop, ή τη χρήση ορισμένων προηγμένων χαρακτηριστικών του ActionScript 3.0, όπως η πρόσβαση σε βάσεις δεδομένων SQLite.

Σε ορισμένες περιπτώσεις, αυτό μπορεί να είναι ένα τεράστιο πρόβλημα, αλλά αν θέλετε απλά να εγκαταστήσετε υπάρχουσες web εφαρμογές ή ταινίες για desktop εφαρμογές χωρίς καμία επιπλέον εργασία ανάπτυξης, τότε το AIR είναι κατάλληλο για τις περισσότερες ανάγκες.

Παρά το γεγονός ότι ο προγραμματισμός AIR επικεντρώνεται στην δημιουργία ActionScript 3.0 εφαρμογών, με τη χρήση του Flash ή του Flex είναι επίσης δυνατό να δημιουργηθούν εφαρμογές που στηρίζονται εξ ολοκλήρου στην HTML.

Το AIR υποστηρίζει πλήρως HTML-based εφαρμογές που χρησιμοποιούν τη διάταξη CSS και JavaScript. Επιπλέον, μπορείτε να χρησιμοποιήσετε άλλα JavaScript Frameworks όπως Prototype, jQuery, ή Mootools προκειμένου να δημιουργήσετε εφαρμογές.

Η διαδικασία για την κατασκευή μιας εφαρμογής γύρω σε HTML είναι το ίδιο σαν να επρόκειτο να υπάρχει ένα αρχείο SWF.

Εδώ, στο content property, καθορίζετε απλά ένα αρχείο HTML που υπάρχει μέσα σε ένα πακέτο εφαρμογής AIR:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.0.M6">
  <id>HTMLApp</id>
  <filename>HTMLApp</filename>
  <version>v1</version>
  <initialWindow>
    <content>app.html </content>
  </initialWindow>
</application>
```

Μέσα σε μια HTML-based εφαρμογή, ένας προγραμματιστής έχει πλήρη πρόσβαση στο AIR και ActionScript 3.0 APIs με το αντικείμενο window.runtime JavaScript. Για παράδειγμα, αν θέλετε να αποκτήσετε πρόσβαση στο URLMonitor τάξη, εσείς θα το πράξουν με τον ίδιο τρόπο όπως θα κάνατε σε ActionScript, παρά μόνον μέσω των window.runtime αντικείμενο στο HTML DOM:

```
var request = new window.runtime.flash.net.URLRequest(
"http://www.foo.com" );
monitor = new window.runtime.air.net.URLMonitor( request );
monitor.addEventListener(
window.runtime.flash.events.StatusEvent.STATUS,
doStatus );
monitor.start();
```

3.6.2.3 Προσθέτοντας εικόνες και οι φακέλους

Εκτός από την προσαρμογή του παραθύρου της εφαρμογής σας, μπορείτε επίσης να ρυθμίσετε και άλλα στοιχεία για την οπτική και μη οπτική εμφάνισή της.

Για να προσθέσετε ένα δικό σας εικονίδιο, μπορείτε να χρησιμοποιήσετε το ακόλουθο κομμάτι XML, το οποίο θα προσθέσετε στο descriptor XML αρχείο, στο <application> κόμβο:

```
<icon>
  <image16x16>assets/icon-16x16.png</image16x16>
  <image32x32>assets/icon-32x32.png</image32x32>
```

```
<image48x48>assets/icon-48x48.png</image48x48>
<image128x128>assets/icon-128x128.png</image128x128>
</icon>
```

Εδώ θα εισάγετε PNG αρχεία που περιέχονται μέσα στο AIR πακέτου της εφαρμογής σας. Αυτά μπορεί να δημιουργηθούν με το Adobe Photoshop ή κάποιο άλλο εργαλείο και θα πρέπει να είναι μεγέθους είτε 16 x 16, 32 x 32, 48 x 48, ή 128 x 128.

Κατά την κατασκευή της εφαρμογής σας, σε αυτή την περίπτωση, θα δημιουργήσετε ένα assets κατάλογο μέσα στο src φάκελο στον οποίο θα τοποθετηθούν οι PNG εικόνες σας. Κατά τη διάρκεια της διαδικασίας compile της εφαρμογής, αυτά τα αρχεία θα πρέπει να γίνονται package αυτόματα.

Μπορείτε επίσης να προσαρμόσετε διάφορα αντικείμενα που χρησιμοποιούνται όταν η εφαρμογή έχει εγκατασταθεί, όπως ο προεπιλεγμένος φάκελος εγκατάστασης και τον φάκελο στον οποίο η συντόμευση του προγράμματος τοποθετείται στο Windows Start menu:

```
<installFolder></installFolder>
<programMenuFolder>Twitter</programMenuFolder>
```

3.7 Διανομή μιας AIR εφαρμογής

Τώρα που έχετε μετατρέψει επιτυχώς το Twitter mashup σας σε μια εφαρμογή AIR, αξίζει τον κόπο να αφιερώσετε λίγο χρόνο να δείτε πώς να εξαγάγετε μια επίσημη έκδοση που να μπορείτε να αρχίσετε να μοιράζεστε με τον κόσμο.

3.7.1 Εγκατάσταση AIR εφαρμογής από την οπτική γωνία του χρήστη

Όταν μια εφαρμογή AIR διανέμεται, διανέμεται με τη μορφή ενός αρχείου .air, το οποίο βασίζεται στην μορφή αρχείου .zip. Τα compiler packages μαζεύουν τα περιεχόμενα της εφαρμογής σας σε ένα φακέλο για την απλούστευση της διανομής. Αυτό το αρχείο είναι μια ανεξάρτητη πλατφόρμα, ώστε το ίδιο αρχείο .air να μπορεί να που χρησιμοποιηθεί σε Mac ή PC. Επιπλέον, προκειμένου να αποτραπεί η παραποίηση του περιεχομένου του, μπορεί να υπογραφεί ψηφιακά από τον δημιουργό.

Η επέκταση αρχείου .air είναι registered σε AIR όταν το AIR είναι εγκατεστημένο αρχικά, για να έχει έτσι ο χρήστης μια έκδοση του AIR εγκατεστημένη στον υπολογιστή του, μπορεί απλά να κάνει DoubleClick ένα αρχείο AIR, για να ξεκινήσει η διαδικασία εγκατάστασης.

Όταν κάνετε διπλό κλικ στο αρχείο AIR, θα εμφανιστεί ένα παράθυρο παρόμοιο με αυτό στο σχήμα, δείχνοντας σαφώς το όνομα της εφαρμογής και του εκδότη, λεπτομέρειες σχετικά με το κατά πόσον η αίτηση έχει ψηφιακή υπογραφή και το επίπεδο της πρόσβασης της εφαρμογής στο σύστημα.

Αυτό εξυπηρετεί ως μια ειδοποίηση προς στους χρήστες ότι κάτι εγκαθίσταται στον υπολογιστή τους. Επειδή οι AIR εφαρμογές μπορούν να έχουν πρόσβαση στο τοπικό σύστημα αρχείων, θα πρέπει να εμφανίζονται με τον ίδιο τρόπο που μια οποιοδήποτε άλλη απλή εφαρμογή θα εμφανιστεί. Η κοινή λογική λέει

να ενημερώνονται οι χρήστες για untrusted εφαρμογές οι οποίες μπορεί να είναι επιβλαβής, και το ίδιο ισχύει και για τις AIR εφαρμογές.

Εντός του-app.xml αρχείου μπορείτε να ορίσετε κάποιες από τις πληροφορίες που θα εμφανίζονται κατά την εγκατάσταση, όπως είναι το όνομα της εφαρμογής και τις λεπτομέρειες εκδότη:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.0.M6">
  <id>SimpleTwitter</id>
  <filename>SimpleTwitter</filename>
  <name>My First Twitter Application</name>
  <version>2</version>
  <description>
    An application which that demonstrates the capabilities of Flex,
    reading a Twitter Feed.
  </description>
  <copyright>Chris Korhonen, 2008.</copyright>

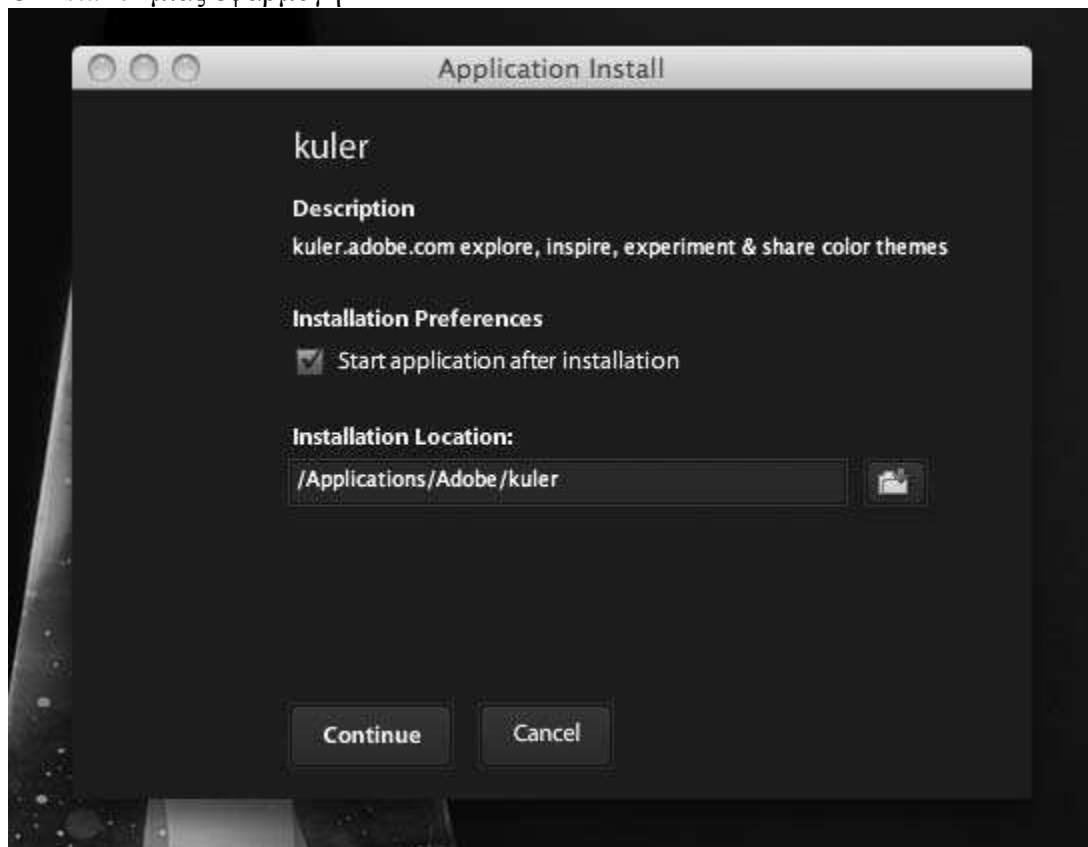
  <initialWindow>
    <content>
      [This value will be overwritten by Flex Builder in the
      output app.xml]
    </content>
    <title>My Twitter Application</title>
  </initialWindow>
</application>
```

Σε αυτό το σημείο, ο χρήστης μπορεί να επιλέξει να γίνει εγκατάσταση ή να ακυρώσει τη διαδικασία.

Εάν ο χρήστης επιλέξει να συνεχίσει την εγκατάσταση, καλείται να επιλέξετε ένα path εγκατάστασης. Στα Windows η προεπιλογή είναι στον κατάλογο Program Files, και σε Mac OS X στο Application Folder. Ως προγραμματιστής, αυτό δεν έχει σημασία, διότι τα AIR APIs παρέχουν μια μέθοδο για την προσπέλαση αρχείων ανεξάρτητα από το λειτουργικό σύστημα.



Ο installer μιας εφαρμογή AIR



Ρύθμιση των επιλογών εγκατάστασης

Κατά τη διαδικασία εγκατάστασης, το AIR δημιουργεί ένα κωδικό για κάθε συγκεκριμένη πλατφόρμα που επιτρέπει στην εγκατεστημένη εφαρμογή να συμπεριφέρεται και να εκτελείται με τον ίδιο τρόπο όπως μια άλλη απλή εφαρμογή. Αυτό είναι το μόνο σημείο που παράγεται κωδικός, και η διαδικασία είναι εντελώς διαφανής για τον προγραμματιστή.

3.7.2 Δημιουργία AIR αρχείου

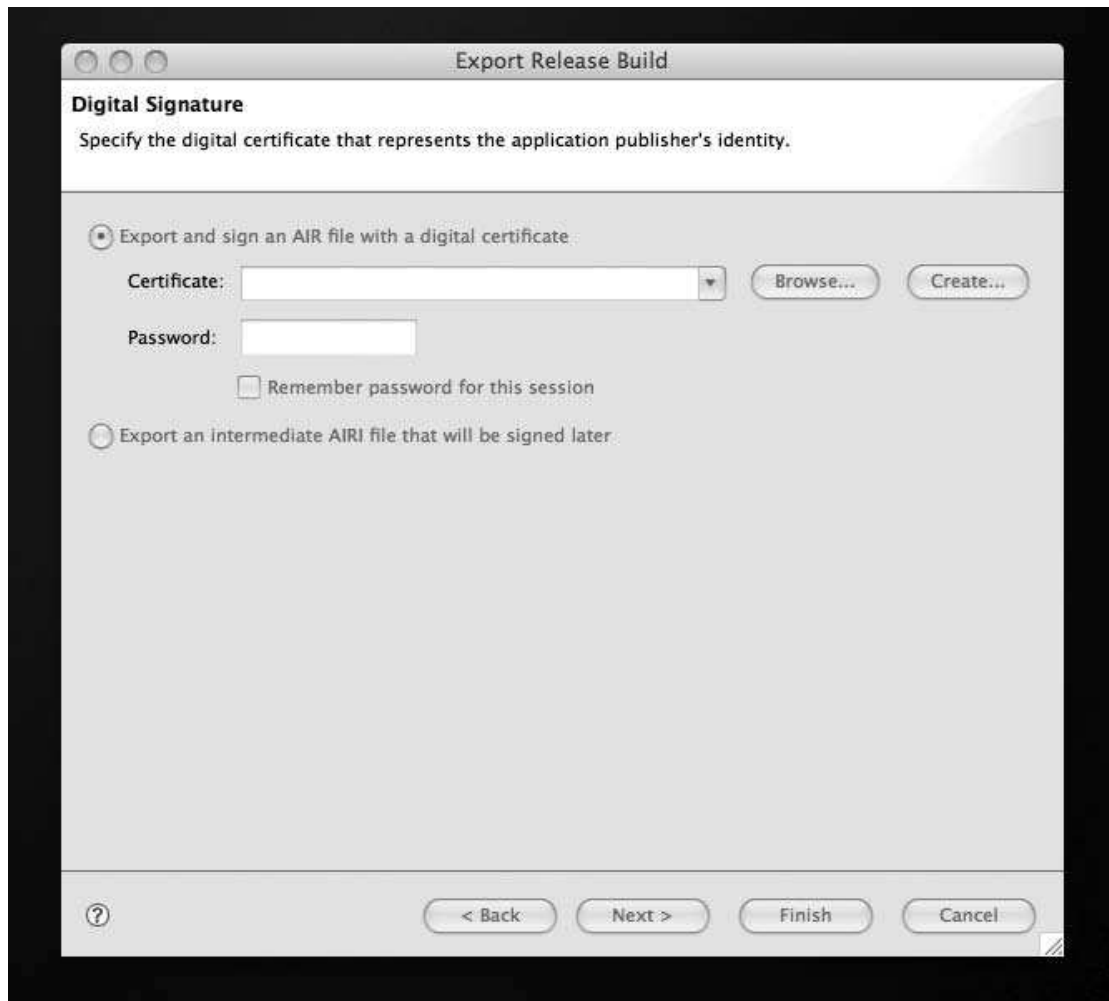
Για να δημιουργήσετε ένα αρχείο AIR για την εφαρμογή σας, ανοίξτε το μενού Project, και επιλέξτε την επιλογή Export Release Build. Αυτό ανοίγει τον οδηγό για την εξαγωγή.



Εξάγοντας ένα release build

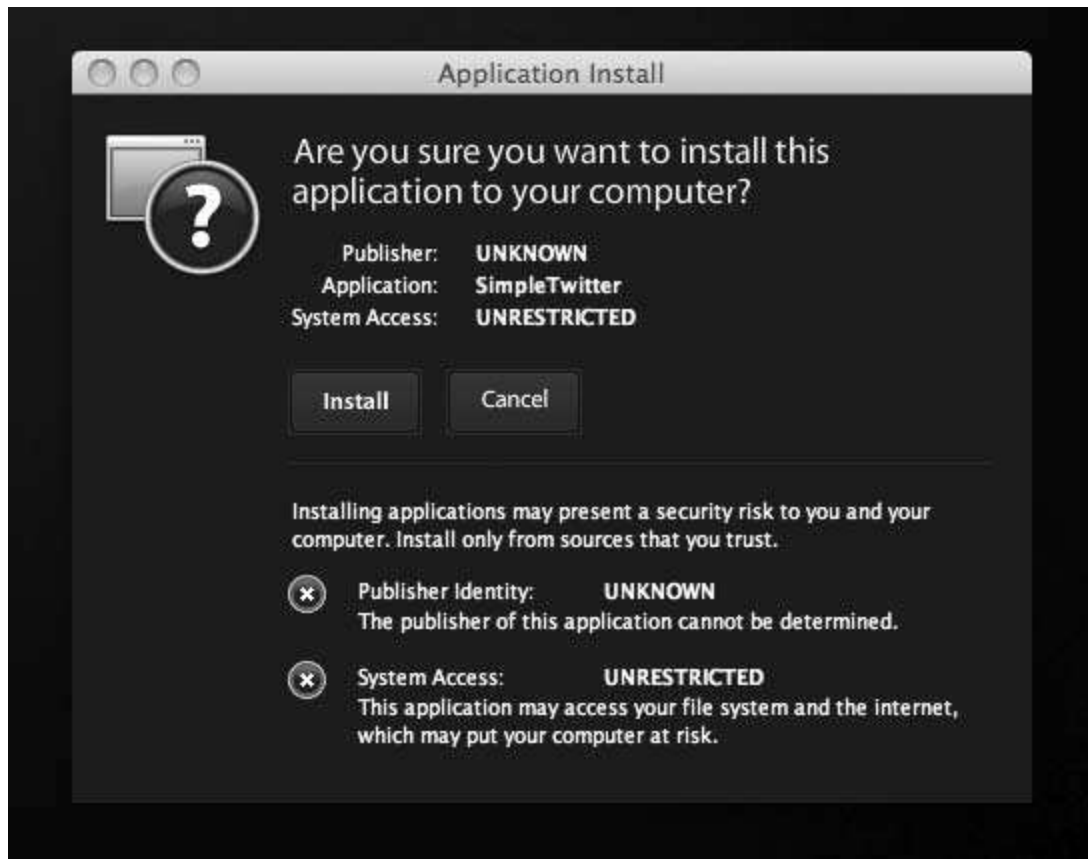
Εδώ μπορείτε να επιλέξετε το project και το MXML αρχείο που θέλετε να εξαγάγετε, καθώς και να προσδιορίσετε τη διαδρομή όπου το AIR αρχείο σας θα δημιουργηθεί.

Στο επόμενο βήμα, θα σας ζητηθεί αν θέλετε να υπογράψετε ψηφιακά το AIR αρχείο σας. Σε γενικές γραμμές αυτό συνιστάται, επειδή δίνει στην εφαρμογή κάποιο βαθμό νομιμότητας και επειδή η παρουσία ενός πιστοποιητικού χρησιμεύει για να καθησυχάσει τον τελικό χρήστη κατά τη διαδικασία εγκατάστασης.



Η ψηφιακή υπογραφή ενός αρχείου AIR

Εάν επιλέξουν να μην υπογράψετε το AIR αρχείο σας δεν είναι δύσκολο να φανταστεί κανείς ότι το μεγάλο κόκκινο ερωτηματικό και τα Xs θέτουν υπό αμφισβήτηση τη νομιμότητα της εφαρμογής και προτρέπουν τον χρήστη να ακυρώσει την εγκατάσταση.



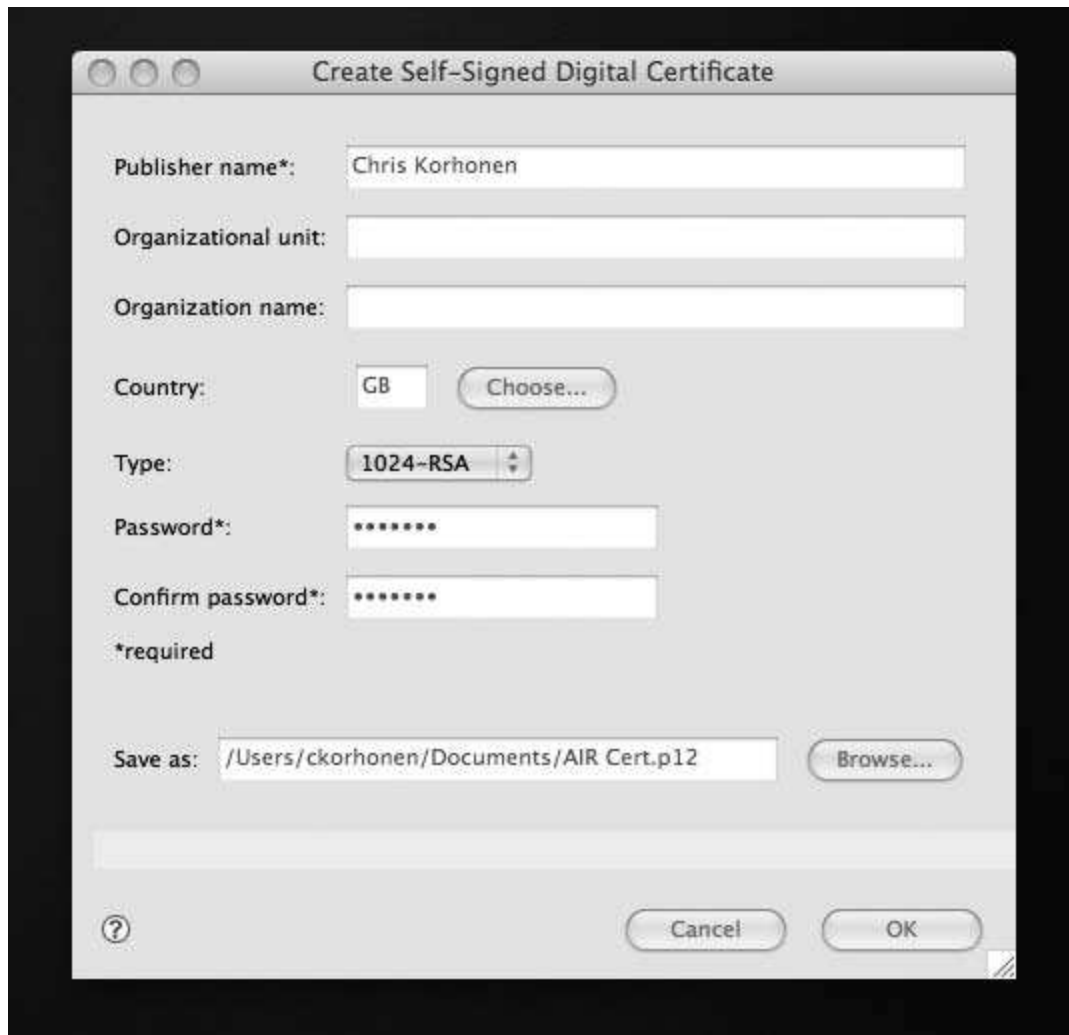
Η διαδικασία εγκατάστασης για ένα ανυπόγραφο αρχείο AIR

Οι εφαρμογές μπορούν να υπογράφονται με ένα πιστοποιητικό, όπως αυτές που παρέχονται από τους εκδότες πιστοποιητικό ασφαλείας, όπως η Thawte (www.thawte.com/) και η VeriSign (www.verisign.com/). Ωστόσο, εάν θέλετε να αποφύγετε την δαπάνη, μπορείτε να δημιουργήσετε το δικό σας πιστοποιητικό μέσα στον Flex Builder.

Για να δημιουργήσετε το δικό σας πιστοποιητικό, κάντε κλικ στο κουμπί Create. Ένα παράθυρο θα ανοίξει και σας ζητά να εισάγετε τις λεπτομέρειες που θέλετε να εμφανίζονται στο ψηφιακό πιστοποιητικό σας. Αυτές μπορεί να περιλαμβάνουν το όνομά σας, το όνομα του οργανισμού σας, το τμήμα σας, και τη γεωγραφική θέση σας.

Μπορείτε επίσης να επιλέξετε το είδος του πιστοποιητικού που θέλετε να δημιουργήσετε. Εδώ είναι ασφαλές να επιλέξετε την προεπιλογή 1024-RSA.

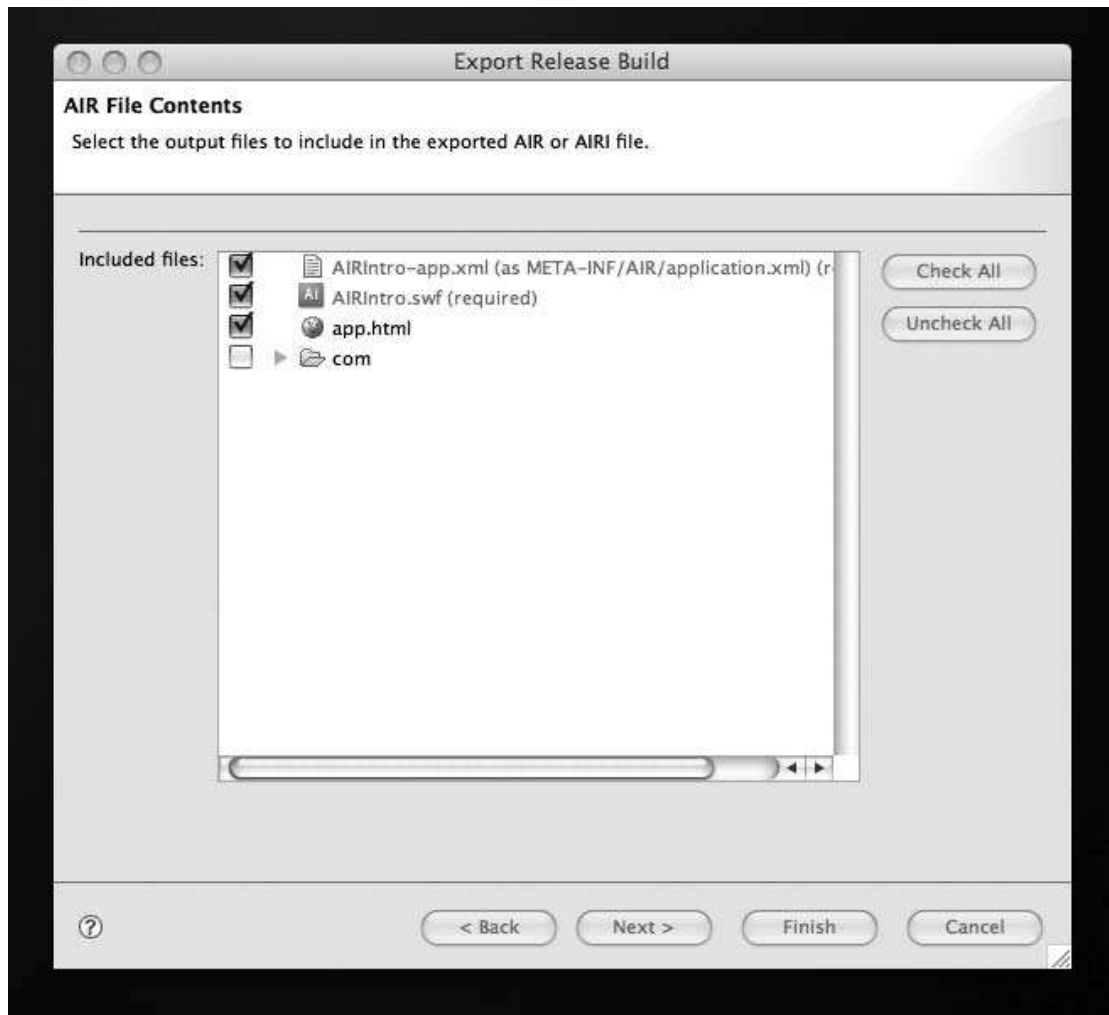
Πρέπει επίσης να θέσετε έναν κωδικό πρόσβασης για το πιστοποιητικό που θα απαιτείται κάθε φορά που θα κάνετε sign in στην εφαρμογή σας.



Δημιουργώντας ένα ψηφιακό πιστοποιητικό

Μόλις ολοκληρωθεί, κάντε κλικ στο OK και το certificate σας θα δημιουργηθεί. Τότε θα είστε σε θέση να επιλέγετε από το drop-down menu πιστοποιητικών και να το χρησιμοποιείτε για να υπογράφετε τις εφαρμογές.

Το τελευταίο μέρος στη διαδικασία εξαγωγής, είναι να επιλέξετε τα αρχεία που θέλετε να συμπεριλάβετε στο δικό σας AIR πακέτο. Από προεπιλογή, περιλαμβάνονται τα περιεχόμενα του καταλόγου bin. Ωστόσο, για την ασφάλεια και την διατήρηση μικρού μεγέθους του αρχείου, είναι λογικό να αποκλείσετε οποιαδήποτε αρχεία ξέρετε ότι δεν χρησιμοποιούνται από το εφαρμογή.



Προσθαφαίρεση αρχείων από το πακέτο AIR

Τώρα θα πρέπει να έχετε ένα αρχείο AIR που θα εγκαταστήσει την εφαρμογή σας.

ΚΕΦΑΛΑΙΟ 4 Ανάλυση Απαιτήσεων Συστήματος

4.1 Εισαγωγή

4.1.1 Τι είναι Google Maps API για Flash;

Αυτό το API δίνει την δυνατότητα στους προγραμματιστές Flex να εισάγουν Google Maps σε Flash εφαρμογές. Όπως στην Javascript έκδοση αυτό το ActionScript API προσφέρει διάφορα εργαλεία για διαχείριση και προσθήκη περιεχομένου σε χάρτες μέσα από μια πληθώρα από υπηρεσίες που επιτρέπουν να εισάγουμε διαδραστικές εφαρμογές με χάρτες στις ιστοσελίδες μας

4.1.2 Εγγραφή για Google Maps API Key

Ο κάθε χρήστης που πρόκειται να χρησιμοποιήσει το Google Maps API στις εφαρμογές του θα πρέπει αρχικά να κάνει εγγραφή στην ιστοσελίδα της Google για να δεχτεί ένα API κλειδί το οποίο θα πρέπει να συμπεριλάβει στις εφαρμογές του.

<http://code.google.com/intl/en/apis/maps/signup.html>

4.1.3 Κατέβασμα του Google Maps API για Flash SDK.

Ο χρήστης στη συνέχεια θα πρέπει να κατεβάσει το [SDK](#) το οποίο είναι ένα .zip αρχείο που περιέχει στον υποφάκελο lib το αρχείο map_flex_1_9.swc αρχείο το οποίο είναι απαραίτητο γιατί περιέχει τους Google Maps που θα χρησιμοποιηθούν από το Flex Builder πρόγραμμα για τη δημιουργία του χάρτη.

4.1.4 Κατασκευή Απλού χάρτη

Στον κώδικα αυτό βλέπουμε πως μπορούμε να κατασκευάσουμε ένα απλό χάρτη.

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
  <maps:Map xmlns:maps="com.google.maps.*"
  mapevent_mappreinitialize="onMapPreinitialize(event)"
  id="map" width="100%" height="100%" key="your_api_key"/>
  <mx:Script>
    <![CDATA[

import com.google.maps.LatLng;
import com.google.maps.Map;
import com.google.maps.MapEvent;
import com.google.maps.MapType;
import com.google.maps.MapOptions;

private function onMapPreinitialize(event:Event):void {
```



```

var myMapOptions:MapOptions = new MapOptions();
myMapOptions.zoom = 14;
myMapOptions.center = new LatLng(40.736072,-73.992062);
myMapOptions.mapType = MapType.NORMAL_MAP_TYPE;
this.map.setInitOptions(myMapOptions);
}

]]>
</mx:Script>
</mx:Application>

```

4.1.5 Προσθήκη του API Key

Για να λειτουργήσει ο χάρτης θα πρέπει να εισάγουμε ένα API key το οποίο μπορούμε να το λάβουμε κάνοντας sign in στην διεύθυνση <http://code.google.com/apis/maps/signup.html> εισάγοντας το προσωπικό μας domain. Στη συνέχεια το εισάγουμε στον κώδικα του χάρτη μας:

```

id="map" width="100%" height="100%" key="
ABQIAAAAW2AtpU_LPdsOHL7TNgdYOBQqF5vs-
amCUieOBLhZ3cTA_oZXRR_RINgXgB3XSe7tpa1A4OSInykUg"/>

```

4.2 Events

Η ActionScript εντός των αρχείων SWF οδηγείται από τα events, δηλαδή η ActionScript ανταποκρίνεται στις αλληλεπιδράσεις δημιουργώντας events, και περιμένει το πρόγραμμα να ανταποκριθεί σε αυτά τα events. Για παράδειγμα, μέσα σε ένα αρχείο Flash, ο χρήστης το ποντίκι και το πληκτρολόγιο δημιουργούν events που διαδίδονται σε όλο το Flash UI. Events μπορούν επίσης να παραχθούν εσωτερικά από το σύστημα, όταν μια αλλαγή κατάστασης εμφανίζεται. Προγράμματα που αντιδρούν σε ορισμένα events δημιουργούν ActionScript Event Listeners για τα events αυτά και εκτελούν κώδικα όταν παραλαμβάνονται.

4.1.2 Event Listeners

Τα events στο Google Maps API για Flash ορίζονται σε συγκεκριμένες κλάσεις (όπως MapEvent). Ορισμένα από αυτά τα events είναι UI events, ενώ άλλα σχετίζονται με τις αλλαγές της κατάστασης του περιβάλλοντος Maps API. Μερικά events σχεδιάστηκαν με ένα συγκεκριμένο αντικείμενο στο μυαλό. Για παράδειγμα, ένα event MapEvent.MAP_READY ενεργοποιείται όταν ο χάρτης είναι έτοιμος να λάβει εντολές. Το event αυτό θα πρέπει να το διαχειρίζεται μόνο ένα αντικείμενο χάρτη.

Άλλα event εφαρμόζονται σε μια μεγάλη ποικιλία αντικειμένων. Για παράδειγμα, το MapMouseEvent CLICK, DOUBLE_CLICK, και DRAG_STEP είναι events που μπορεί να είναι ενσωματωμένα map objects ή overlays, και μπορεί να απαιτούν διαφορετικές ενέργειες, ανάλογα με το αντικείμενο που λαμβάνει το event.

Για παράδειγμα χρησιμοποιήστε την Map.addListener () μέθοδο. Η μέθοδος αυτή λαμβάνει ως ορίσματα ένα όνομα event και μια function για να

καλέσετε όταν συμβαίνει το συγκεκριμένο event. Για παράδειγμα, αυτός ο κώδικας εμφανίζει ένα InfoWindow κάθε φορά που ο χρήστης κάνει κλικ πάνω στον χάρτη:

```
private function onMapReady(event:MapEvent):void {
    map.setCenter(new LatLng(37.4419, -122.1419), 13,
    MapType.NORMAL_MAP_TYPE);
    map.addListener(MapMouseEvent.CLICK, onMapClick);
}

private function onMapClick(event:MapMouseEvent):void {
    map.openInfoWindow(event.latLng,
    new InfoWindowOptions({title: "Click Event", content: "You clicked the map!"}));
}
```

4.3 Controls

Οι χάρτες <http://maps.google.com> περιέχουν UI στοιχεία που επιτρέπουν την αλληλεπίδραση του χρήστη με τον χάρτη. Αυτά τα στοιχεία είναι γνωστά ως controls και μπορείτε να τα συμπεριλάβετε στο Google Maps API σας για την εφαρμογή Flash.

4.3.1 Προσθήκη Controls στον χάρτη

Μπορείτε να Προσθέσετε τα controls στον χάρτη με την μέθοδο `addControl()`. Για παράδειγμα, για να προσθέσετε το panning control που βλέπετε στους Χάρτες Google στο χάρτη σας, θα πρέπει να εισάγετε την ακόλουθη γραμμή στο χάρτη σας:

```
map.addControl(new PositionControl());
```

Μπορείτε να προσθέσετε διάφορα controls σε έναν χάρτη. Σε αυτή την περίπτωση, προσθέτουμε το `ZoomControl`, `PositionControl`, και `MapTypeControl`, η οποία μας επιτρέπουν να κάνουμε ζουμ στο χάρτη και επιλέξουμε διαφορετικών τύπων χάρτη. Τα control είναι σε πλήρη λειτουργία από τη στιγμή που περιλαμβάνονται σε ένα χάρτη.

```
private function onMapReady(event:MapEvent):void {
    map.setCenter(new LatLng(42.366662,-71.106262), 11,
```

```

MapType.NORMAL_MAP_TYPE);
map.addControl(new ZoomControl());
map.addControl(new PositionControl());
map.addControl(new MapTypeControl());
}

```



4.3.2 Ρύθμιση Controls

Ορισμένα controls (όπως το MapTypeControl) είναι διαμορφώσιμα. Για παράδειγμα, από προεπιλογή, το Google Maps API προβλέπει τέσσερις μορφές χάρτη: NORMAL_MAP_TYPE, SATELLITE_MAP_TYPE, HYBRID_MAP_TYPE, και PHYSICAL_MAP_TYPE. Μπορείτε να μεταβάλετε το είδος χάρτη που διατίθεται σε ένα χάρτη με την διαγραφή των υφιστάμενων τύπων μέσω Map.removeMapType () ή την προσθήκη μέσω Map.addMapType (). Ο παρακάτω κώδικας αφαιρεί το HYBRID_MAP_TYPE από τους διαθέσιμους τύπους χάρτη, αφήνοντας μόνο τρεις τύπους χάρτη. Αφού προσθέσουμε το MapTypeControl, μόνο αυτοί οι τρεις τύποι είναι διαθέσιμοι.

```

private function onMapReady(event:MapEvent):void {
map.setCenter(new LatLng(42.366662,-71.106262), 11,
MapType.NORMAL_MAP_TYPE);
map.removeMapType(MapType.HYBRID_MAP_TYPE);
map.addControl(new MapTypeControl());
}

```

4.3.3 Τοποθέτηση Controls στο χάρτη

Η `setControlPosition()` μέθοδος λαμβάνει μια παράμετρο `ControlPosition` που σας επιτρέπει να προσδιορίσετε τη θέση του control στο χάρτη σας. Η τιμή αυτή μπορεί να είναι μία από τις ακόλουθες τιμές, όπου η κάθε προσδιορίζει μια γωνία του χάρτη στον οποίο θα τοποθετήσετε το control:

```
ANCHOR_TOP_RIGHT  
ANCHOR_TOP_LEFT  
ANCHOR_BOTTOM_RIGHT  
ANCHOR_BOTTOM_LEFT
```

Η `ControlPosition` μπορεί προαιρετικά να καθορίσει ένα offset που υποδηλώνει σε πόσα pixels από την άκρη του χάρτη θα τοποθετηθεί το control χρησιμοποιώντας x, y τιμές όπου x ένα οριζόντιο offset ενώ y ένα κάθετο.

Στο παρακάτω παράδειγμα προσθέτουμε τη `MapTypeControl` στην πάνω δεξιά γωνία του χάρτη με 16 pixels οριζόντια και 10 pixels κάθετα.

```
private function onMapReady(event:MapEvent):void {  
    map.setCenter(new LatLng(37.4419, -122.1419), 13,  
    MapType.NORMAL_MAP_TYPE);  
  
    var bottomRight:ControlPosition = new  
    ControlPosition(ControlPosition.ANCHOR_BOTTOM_RIGHT, 16, 10);  
    var myMapTypeControl:MapTypeControl = new MapTypeControl();  
    myMapTypeControl.setControlPosition(bottomRight);  
    map.addControl(myMapTypeControl);  
}
```

4.4 Overlays

Overlays είναι αντικείμενα στο χάρτη που συνδέονται με γεωγραφικού πλάτους / μήκους συντεταγμένες, έτσι ώστε να κινούνται όταν σύρετε ή να μεγεθύνετε το χάρτη. Τα overlays είναι τα αντικείμενα που προσθέτετε στο χάρτη για να ορίσετε σημεία, γραμμές, ή περιοχές.

Το Maps API έχει πολλούς τύπους overlays:

Σημεία πάνω στον χάρτη που εμφανίζονται με markers, και συχνά εμφανίζουν μια εικόνα. Markers είναι αντικείμενα τύπου `Marker` και μπορούν να χρησιμοποιήσουν custom εικόνες ως icons.

Lines στο χάρτη (που αντιπροσωπεύουν μια συλλογή από τα σημεία). Τα Lines είναι αντικείμενα τύπου `Polyline`.

Areas στο χάρτη που εμφανίζονται είτε ως polygons, εάν το σχήμα τους πάνω στο χάρτη είναι τυχαίο, είτε ως ground overlays αν είναι ορθογώνια.

Τα InfoWindows είναι επίσης ένα ιδιαίτερο είδος overlay. Σημειώστε, ωστόσο, ότι το InfoWindow προστίθεται στο χάρτη αυτόματα, και ότι μπορεί να υπάρχει μόνο ένα αντικείμενο του τύπου InfoWindow να συνδέεται με ένα χάρτη.

Κάθε overlay επεκτείνει το IOverlay interface. Τα Overlays μπορούν να προστεθούν σε ένα χάρτη με την Map.addOverlay () μέθοδο και να αφαιρεθούν χρησιμοποιώντας την Map.removeOverlay () μέθοδο. (Σημειώστε ότι το InfoWindow, προστίθεται από προεπιλογή στον χάρτη, αν και δεν είναι αρχικά ορατό.)

4.4.1 Markers

Τα Markers προσδιορίζουν τα σημεία στο χάρτη. Ο Marker constructor λαμβάνει ένα LatLng και ένα MarkerOptions αντικείμενο ως arguments. Αυτά τα MarkerOptions αντικείμενα σας επιτρέπουν να αλλάξετε τις default επιλογές των Markers.

Τα Markers σχεδιαστεί για να είναι διαδραστικά. Από προεπιλογή, λαμβάνουν για παράδειγμα MapMouseEvent.CLICK και συχνά χρησιμοποιούνται μέσα σε Event Listeners για να εμφανίσουν InfoWindows.

```
private function onMapReady(event:Event):void {
    map.setCenter(new LatLng(37.4419, -122.1419), 13,
    MapType.NORMAL_MAP_TYPE);

    var markerA:Marker = new Marker(
        new LatLng(48.858842, 2.346997),
        new MarkerOptions({
            strokeStyle: new StrokeStyle({color: 0x987654}),
            fillStyle: new FillStyle({color: 0x223344, alpha: 0.8}),
            radius: 12,
            hasShadow: true
        }));
    map.addOverlay(markerA);
}
```



4.4.2 Polygons

Τα Πολύγωνα είναι παρόμοια με τα Polyline αντικείμενα που αποτελούνται από μια σειρά σημείων. Όπως και στα polylines, μπορείτε να ορίσετε χρώματα, πάχη γραμμών, καθώς και θολερότητα για τις άκρες του πολυγώνου όπως και για την περιοχή μέσα στο πολύγωνο. Τα χρώματα θα πρέπει να έχουν μια αριθμητική τιμή από το `com.google.maps.Color`.

Το ακόλουθο κομμάτι κώδικα δημιουργεί ένα κουτί 10-pixel γύρω από τέσσερα σημεία.

```
private function onMapReady(event:MapEvent):void {
    map.setCenter(new LatLng(37.4419, -122.1419), 13,
    MapType.NORMAL_MAP_TYPE);
    map.addControl(new ZoomControl());
    var latLng:LatLng = map.getCenter();
    var lat:Number = latLng.lat();
    var lon:Number = latLng.lng();
    var latOffset:Number = 0.01;
    var lonOffset:Number = 0.01;
    var polygon:Polygon = new Polygon([
        new LatLng(lat, lon - lonOffset),
        new LatLng(lat + latOffset, lon),
        new LatLng(lat, lon + lonOffset),
        new LatLng(lat - latOffset, lon),
        new LatLng(lat, lon - lonOffset)
    ],
    new PolygonOptions({
        strokeStyle: new StrokeStyle({
```

```

    color: 0x0000ff,
    thickness: 10,
    alpha: 0.7}),
  fillStyle: new FillStyle({
    color: 0x0000ff,
    alpha: 0.7})
  }));
  map.addOverlay(polygon);
}

```



4.4.3 Ground Overlays

Τα πολύγωνα είναι χρήσιμα overlays που αντιπροσωπεύουν αυθαίρετου μεγέθους περιοχές, αλλά δεν μπορούν να αναπαράγουν εικόνες. Αν έχετε μια εικόνα που επιθυμείτε να τοποθετήσετε σε ένα χάρτη, μπορείτε να χρησιμοποιήσετε ένα αντικείμενο GroundOverlay. Ο constructor για GroundOverlay παίρνει ένα URL και τα LatLngBounds της εικόνας ως παραμέτρους. Το ακόλουθο παράδειγμα προσθέτει μια εικόνα αντίγραφο του Newark, NJ στο χάρτη ως overlay:

```

private function onMapReady(event:Event):void {
  map.setCenter(new LatLng(40.740, -74.18), 12,
  MapType.NORMAL_MAP_TYPE);
  map.addControl(new ZoomControl());
  map.addControl(new MapTypeControl());

  var testLoader:Loader = new Loader();
}

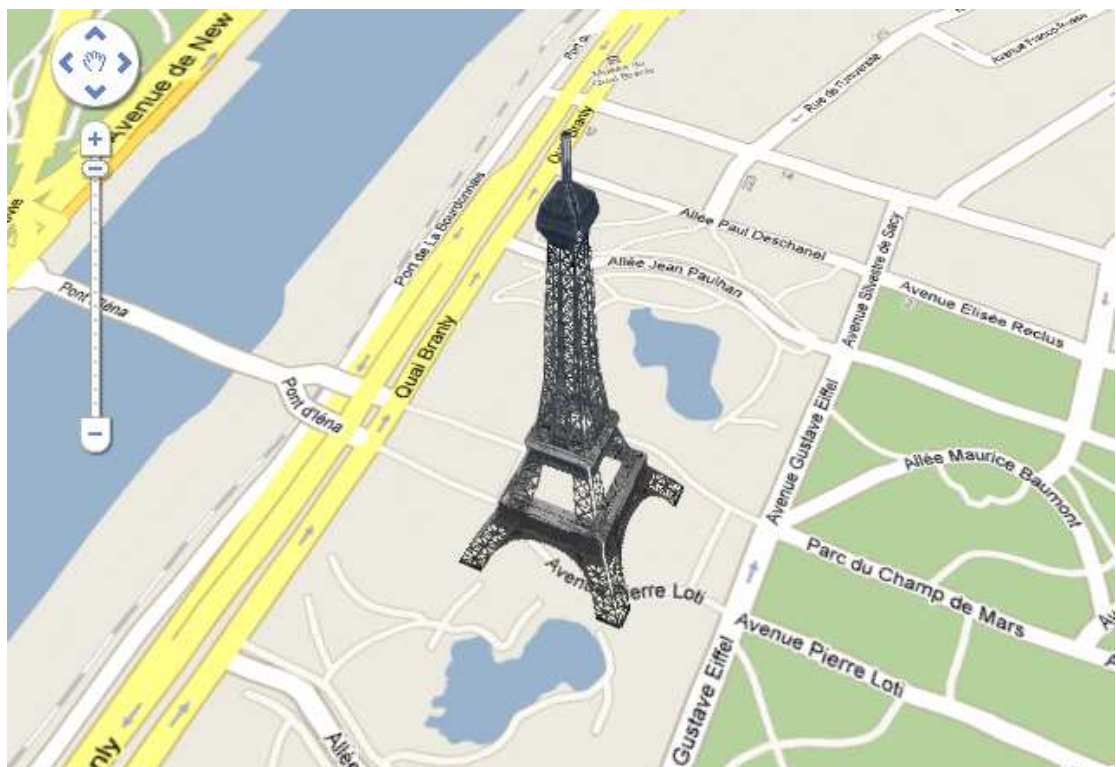
```



```

var urlRequest:URLRequest = new
URLRequest("http://www.lib.utexas.edu/maps/historical/newark_nj_1922.jpg");
testLoader.contentLoaderInfo.addEventListener(Event.COMPLETE,
function(e:Event):void {
    var groundOverlay:GroundOverlay = new GroundOverlay(
        testLoader,
        new LatLngBounds(new LatLng(40.716216,-74.213393), new
LatLng(40.765641,-74.139235)));
    map.addOverlay(groundOverlay);
});
testLoader.load(urlRequest);
}

```



4.4.4 InfoWindows

Κάθε χάρτης στο Google Maps API μπορεί να παρουσιάσει ένα μοναδικό "παράθυρο πληροφοριών" του τύπου InfoWindow, το οποίο εμφανίζει κείμενο ή HTML. Το InfoWindow φαίνεται λίγο σαν ένα μπαλόνι κόμικς. Μπορείτε να δείτε το InfoWindow σε δράση κάνοντας κλικ σε ένα marker στο Google Maps.

Το InfoWindow δεν έχει constructor. Ένα InfoWindow δημιουργείται αυτόματα και επισυνάπτεται στο χάρτη κατά τη δημιουργία του χάρτη. Δεν μπορείτε να δείτε περισσότερα από ένα InfoWindow ταυτόχρονα σε ένα συγκεκριμένο χάρτη, αλλά μπορείτε να μετακινήσετε το InfoWindow και να αλλάξετε το περιεχόμενό του, όπως απαιτείται.

Το Map Object παρέχει μια openInfoWindow () μέθοδο, η οποία παίρνει

LatLng και InfoWindowOptions ως ορίσματα και χρησιμοποιείται για να προσθέσει στο InfoWindow περιεχόμενο. Το παρακάτω παράδειγμα κώδικα είναι ένα InfoWindow στο κέντρο του χάρτη με ένα απλό "Hello, world" μήνυμα.

```
<maps:Map xmlns:maps="com.google.maps.*" id="map"
  mapevent_mapready="onMapReady(event)"
  width="100%" height="100%" language="de" key="your_api_key"/>
```

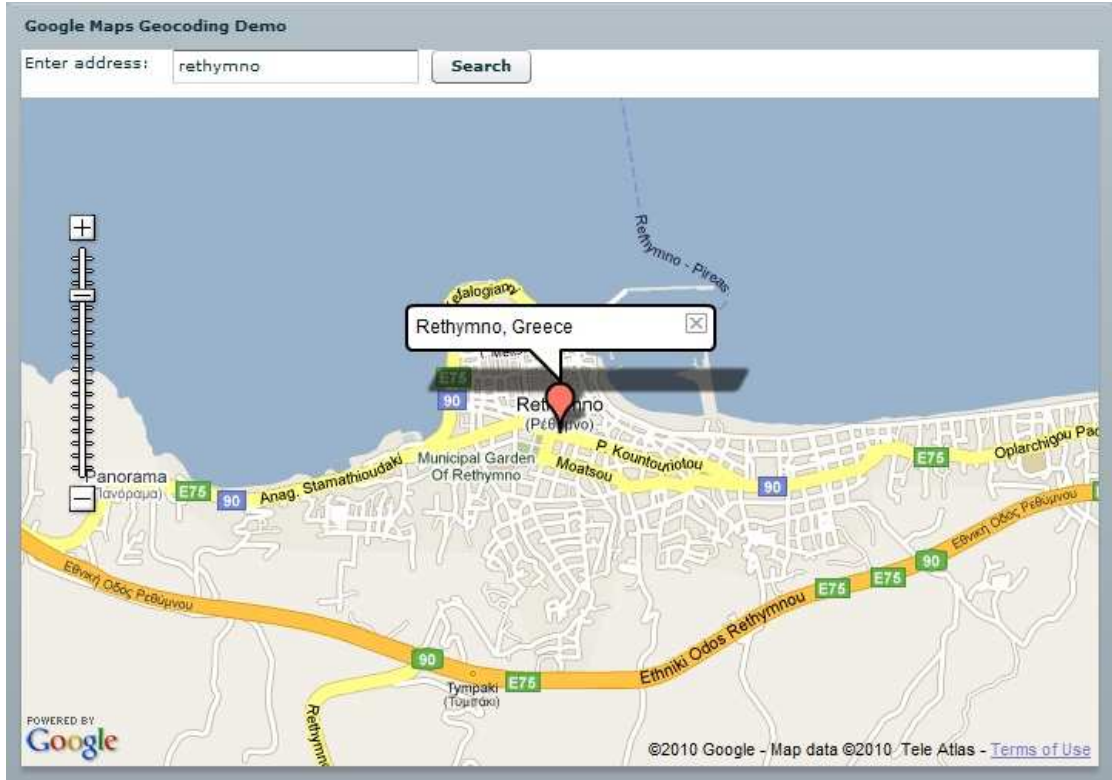


4.5 Services

Το Google Maps API για Flash, επεκτείνεται τακτικά, προσθέτοντας νέες λειτουργίες και χαρακτηριστικά. Τα Service είναι χαρακτηριστικά τα οποία επικοινωνούν με άλλες υπηρεσίες της Google. Ως εκ τούτου, οι υπηρεσίες αυτές είναι συνήθως ασύγχρονες και απαιτούν ειδική αντιμετώπιση.

4.5.1 Geocoding

Geocoding είναι η διαδικασία μετατροπής των διευθύνσεων (όπως το "1600 Amphitheatre Parkway, Mountain View, CA") σε γεωγραφικές συντεταγμένες (όπως γεωγραφικό πλάτος και μήκος 37,423021 -122,083739), το οποίο μπορείτε να χρησιμοποιήσετε για να τοποθετήσετε σημάνσεις ή τη θέση του χάρτη. Το Google Maps API για Flash παρέχει ένα geocoder client για να κάνει geocoding διευθύνσεις δυναμικά από την εισαγωγή δεδομένων από τον χρήστη. Ακόμα αυτό μπορεί να γίνει και στατικά.



ΚΕΦΑΛΑΙΟ 5 Σχεδίαση Συστήματος

5.1 Αρχιτεκτονική Εφαρμογής

Δήλωση εφαρμογής:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application
    xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute">
</mx:Application>
```

Προσθήκη βιβλιοθηκών:

```
import mx.events.ListEvent;
import mx.events.ItemClickEvent;
import mx.controls.advancedDataGridClasses.SortInfo;
import com.google.maps.overlays.MarkerOptions;
import com.google.maps.styles.FillStyle;
import com.google.maps.styles.StrokeStyle;
import com.google.maps.overlays.Marker;
import mx.controls.dataGridClasses.DataGridColumn;
import com.google.maps.interfaces.IMapType;
import com.google.maps.LatLng;
import com.google.maps.Map;
import com.google.maps.MapEvent;
import com.google.maps.MapType;
import com.google.maps.controls.PositionControl;
import com.google.maps.controls.MapTypeControl;
import com.google.maps.controls.ZoomControl;
import com.google.maps.MapMouseEvent;
import com.google.maps.LatLngBounds;
import com.google.maps.MapMouseEvent;
import com.google.maps.InfoWindowOptions;
import com.google.maps.styles.FillStyle;
import mx.controls.Alert;
```

DATASET δεδομένων:

```
private var dataSet : XML = <dataset>
<data location="PAGKRHTIA SYNETAIRISTIKH TRAPEZA(PANCRETA
BANK)" address="Lewforos Ikarou & Efesou"
phone="2810338800" pc="71306" region="Hracleio" county="Hracleio"
lat="35.339074" lon="-25.140476"/> ...
<data location="MILLENIUM BANK" address="A. Papandreou 11"
phone="2821043700" pc="73132" region="Chania" county="Chania"
lat="35.513956" lon="-24.0218913"/>
```

</dataset>

Κλάση categories:

```
private var categories:Object =
{  "visa": {
    "color": 0xFF0000,
    "markers": [],
    "maestro": {
    "color": 0xFF0000,
    "markers": [],
    "diners": {
    "color": 0xFF0000,
    "markers": [],
    "american": {
    "color": 0xFF0000,
    "markers": []}
    };
};
```

Συνάρτηση onContainerInitialize():

```
public function onContainerInitialize( event:Event ):void {
    map = new Map();
    map.key = "ABQIAAAAW2AtpU_LPdsOHL7TNgdYOBQqF5vs-
amCUlieOBLhZ3cTA_oZXRR_RINgXgB3XSe7tpa1A4OSInykUg";
    map.addEventListener( MapEvent.MAP_READY, onMapReady );
    mapContainer.addChild( map );
}
```

Συνάρτηση onContainerResize():

```
public function onContainerResize( event:Event ):void {
    map.setSize( new Point( mapContainer.width, mapContainer.height ) );
}
```

Συνάρτηση onMapReady():

```
private function onMapReady( event:Event ):void {
    map.enableScrollWheelZoom();
    map.enableContinuousZoom();
    map.addControl(new ZoomControl());
    map.addControl(new PositionControl());
    map.addControl(new MapTypeControl());
}
```

```

map.setCenter(new LatLng(35.2908333, 24.8808333), 9,
MapType.NORMAL_MAP_TYPE );
var xmlString:URLRequest = new URLRequest("markers.xml");
var xmlLoader:URLLoader = new URLLoader(xmlString);
xmlLoader.addEventListener("complete", readXml);
}

```

Συνάρτηση *readXml()*:

```

public function readXml(event:Event):void{
    var markersXML:XML = new XML(event.target.data);
    var markers:XMLList = markersXML..marker;
    var markersCount:int = markers.length();
    for (var i:Number = 0; i < markersCount; i++) {
        var marker:XML = markers[i];
        var name:String = marker.@name;
        var address:String = marker.@address;
        var type:String = marker.@type;
        var latlng:LatLng = new LatLng(marker.@lat, marker.@lng);
        createMarker(latlng, name, address, type);
    }
}

```

Συνάρτηση *createMarker()*:

```

public function createMarker(latlng:LatLng, name:String, address:String,
type:String):void {
    var markerOptions:MarkerOptions = new MarkerOptions({ });
    var fillStyle:FillStyle = new FillStyle({ color: categories[type].color});
    markerOptions.fillStyle = fillStyle;
    var marker:Marker = new Marker(latlng, markerOptions);
    var html:String = "<b>" + name + "</b> <br/>" + address;
    marker.addEventListener(MapMouseEvent.CLICK,
function(e:MapMouseEvent):void {
        marker.openInfoWindow(new InfoWindowOptions({ contentHTML:html}));
    });
    categories[type].markers.push(marker);
    map.addOverlay(marker);
}

```

Συνάρτηση *onGridItemClick()*:

```

private function onGridItemClick(event>ListEvent):void{
    var data : XML = event.itemRenderer.data as XML;
    if ( data )
        map.setCenter( new LatLng( parseFloat( data.@lat ), -parseFloat(
data.@lon ) ) );
}

```

```
}
```

Συνάρτηση toggleCategory():

```
private function toggleCategory(type:String):void {  
    for (var i:Number = 0; i < categories[type].markers.length; i++) {  
        var marker:Marker = categories[type].markers[i];  
        if (!marker.visible) {  
            marker.visible = true;  
        } else {  
            marker.visible = false;  
        }  
    }  
}
```

Interface VDividedBox:

```
<mx:VDividedBox width="100%" height="100%">  
</mx:VDividedBox>
```

Interface UIComponent:

```
<mx:UIComponent id="mapContainer"  
    initialize="onContainerInitialize(event);"  
    resize="onContainerResize(event)"  
    width="100%" height="100%" />
```

Interface TabNavigator:

```
<mx:TabNavigator width="100%" height="200">  
</mx:TabNavigator>
```

Interface Canvas:

```
<mx:Canvas width="100%" height="100%" label="BANKS">  
</mx:Canvas>
```

```
<mx:Canvas width="100%" height="100%"  
label="SUPPORT
```



```
</mx:Canvas>
```

Πηγή Δεδομένων DataGrid:

```

<mx:DataGrid dataProvider="{ dataSet.data }" width="100%" height="100%"
itemClick="onGridItemClick(event)" >
  <mx:columns>
    <mx:DataGridColumn headerText="BANK NAME" dataField="@location"
  />
  <mx:DataGridColumn headerText="ADDRESS" dataField="@address" />
  <mx:DataGridColumn headerText="PHONE NUMBER"
dataField="@phone"/>
  <mx:DataGridColumn headerText="POSTAL CODE" dataField="@pc" />
  <mx:DataGridColumn headerText="REGION" dataField="@region" />
  <mx:DataGridColumn headerText="COUNTY" dataField="@county" />
  </mx:columns>
</mx:DataGrid>

```

Interface Panel:

```

<mx:Panel title="" width="100%" height="100%">
</mx:Panel>

```

Interface HDividedBox:

```

<mx:HDividedBox width="100%" height="100%">
</mx:HDividedBox>

```

Interface VBox:

```

<mx:VBox width="20%" height="100%" paddingLeft="5" paddingTop="5">
</mx:VBox>

```

Interface Hbox:

```

<mx:HBox>
  <mx:CheckBox
selected="true"
change="toggleCategory('visa')"/>
  <mx:Label
text="Visa/MasterCard/Cirrus"/>
</mx:HBox>

```

Αρχείο Markers.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<markers>

```

```
<marker name="LAIKH TRAPEZA (ELLAS) AE" address="Idhs 26"
phone="2810285300" pc="71201" region="Hrakleio" county="Hrakleio"
lat="35.3389035" lng="25.1324448" type="visa"/> ...
<marker name="MILLENIUM BANK" address="A. Papandreou 11"
phone="2821043700" pc="73132" region="Chania" county="Chania"
lat="35.513956" lng="24.0218913" type="maestro"/></markers>
```

5.2 Περιγραφή Κλάσεων και συναρτήσεων

Δήλωση εφαρμογής:

Σε αυτό το κομμάτι δηλώνουμε στο Adobe Flex ότι πρόκειται να ακολουθήσει κώδικας για Flex εφαρμογή η οποία μπορεί να τρέχει πάνω σε μια ιστοσελίδα η να γίνει export και να τρέχει αυτόνομα σε adobe flash. Ακόμα δηλώνουμε την έκδοση της XML γλώσσας που θα χρησιμοποιηθεί, το encoding και το layout της εφαρμογής.

Προσθήκη βιβλιοθηκών:

Η προσθήκη βιβλιοθηκών γίνεται με την εντολή import και στην συνέχεια με το όνομα της βιβλιοθήκης που θέλουμε να προσθέσουμε στο πρόγραμμά μας ώστε να είναι σε θέση να εκτελέσει τις εντολές τις οποίες θα ορίσουμε παρακάτω.

DATASET δεδομένων:

Το dataset είναι ένας πίνακας δεδομένων στον οποίο θα συμπεριλάβουμε τα στοιχεία τα οποία θέλουμε να εμφανίζονται αργότερα στο grid της εφαρμογής μας.

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
PAGKRHTIA SYNETAIF	Leforos Ikarou & Efe	2810338800	71306	Hrakleio	Hrakleio
AGROTIKH TRAPEZA	Karamanlh Kwnstanti	2821074572	73135	Chania	Chania
MILLENIUM BANK	L. Portaliou k Papana	2831050773	74100	Rethymno	Rethymno
TRAPEZA PEIRAIWS A	Evans	2810343343	71201	Hrakleio	Hrakleio

Στο πεδίο data location βάζουμε το όνομα της τράπεζας που θέλουμε να προσθέσουμε, στο πεδίο address την διεύθυνση της τράπεζας, στο πεδίο phone το τηλέφωνο επικοινωνίας με την τράπεζα, στο πεδίο pc τον ταχυδρομικό κώδικα της περιοχής στην οποία βρίσκετε η τράπεζα, στο πεδίο region την πόλη η το χωριό και στο πεδίο county τον νομό που βρίσκεται η τράπεζα. Στα πεδία lat και lng βάζουμε τις γεωγραφικές συντεταγμένες στις οποίες βρίσκεται η τράπεζα πάνω στο χάρτη του Google maps.

Κλάση categories:

Στην κλάση `categories` ορίζουμε τις κατηγορίες που θα χρησιμοποιήσουμε στην εφαρμογή μας οι οποίες είναι `visa` (`visa`, `mastercard`, `cirrus`), `maestro`, `diners` (`Diners Club International`) και `American` (`American Express`), καθώς και το χρώμα που θα έχει ο `marker` της κάθε μιας πάνω στον χάρτη. Για χρώμα έχει οριστεί σε όλες το ίδιο `0xFF0000` (κόκκινο).

Συνάρτηση `onContainerInitialize()`:

Με αυτή την συνάρτηση πετυχαίνουμε να μεταβάλλεται το μέγεθος του χάρτη ανάλογα με την μεταβολή του μεγέθους της εφαρμογής και ειδικότερα του `container` στο οποίο έχουμε συμπεριλάβει τον χάρτη μας.

Συνάρτηση `onMapReady()`:

Η μέθοδος `onMapReady` είναι η πρώτη μέθοδος που θα υλοποιηθεί από τη στιγμή που θα τοποθετηθεί ο χάρτης στην εφαρμογή μας. Οι εντολές `map.enableScrollWheelZoom()`; Και `map.enableContinuousZoom()`; Προσθέτουν την δυνατότητα για `zoom` στον χάρτη με την χρήση του `scroll` του ποντικιού. Η εντολή `map.addControl(new ZoomControl());` Προσθέτει `control` στην πάνω αριστερή πλευρά του χάρτη και ο χρήστης έχει από εκεί την δυνατότητα να κάνει `zoom` κάνοντας κλικ στα διάφορα κουμπιά του `control`. Με την εντολή `map.addControl(new PositionControl());` Ο χρήστης έχει την δυνατότητα να μεταφερθεί προς όποια κατεύθυνση θέλει το τρέχον κέντρο του χάρτη. Με την εντολή `map.addControl(new MapTypeControl());` Εμφανίζεται στο πάνω δεξί μέρος του χάρτη ένα `panel` με 4 επιλογές: Χάρτης, δορυφόρος, Υβριδικός και έδαφος. Ο χρήστης μπορεί να διαλέξει οποιαδήποτε από αυτές τις 4 επιλογές προβολής χάρτη θέλει. Με την εντολή `map.setCenter(new LatLng(35.2908333, 24.8808333), 9, MapType.NORMAL_MAP_TYPE);` Ορίζουμε ποιο θα είναι το κέντρο του χάρτη τη στιγμή που θα ανοίξουμε το πρόγραμμα μας. Στην προκειμένη περίπτωση έχουμε ορίσει της συντεταγμένες `35.2908333, 24.8808333` που είναι το γεωγραφικό πλάτος και ύψος που βρίσκεται η Κρήτη στον χάρτη του `google maps`. Ακόμα ο αριθμός `9` δηλώνει το μέγεθος του `zoom` που θέλουμε να έχει ο χάρτη όταν ξεκινήσει η εφαρμογή. Θα ορίσουμε ποιο `type` από τα 4 θέλουμε να έχει από `default` ο χάρτης. Τα `types` είναι: `NORMAL_MAP_TYPE`, `SATELLITE_MAP_TYPE`, `HYBRID_MAP_TYPE`, και `PHYSICAL_MAP_TYPE`. Τέλος δηλώνουμε της μεταβλητές `xmlLoader` και `xmlRequest` οι οποίες θα χρησιμοποιηθούν για να διαβάσει αργότερα το πρόγραμμα τα δεδομένα από το εξωτερικό αρχείο `marker.xml` το οποίο περιέχει τα σημεία τα οποία θα τοποθετηθούν τα `markers` των `τραπεζών` πάνω στον χάρτη.

Συνάρτηση `readXml()`:

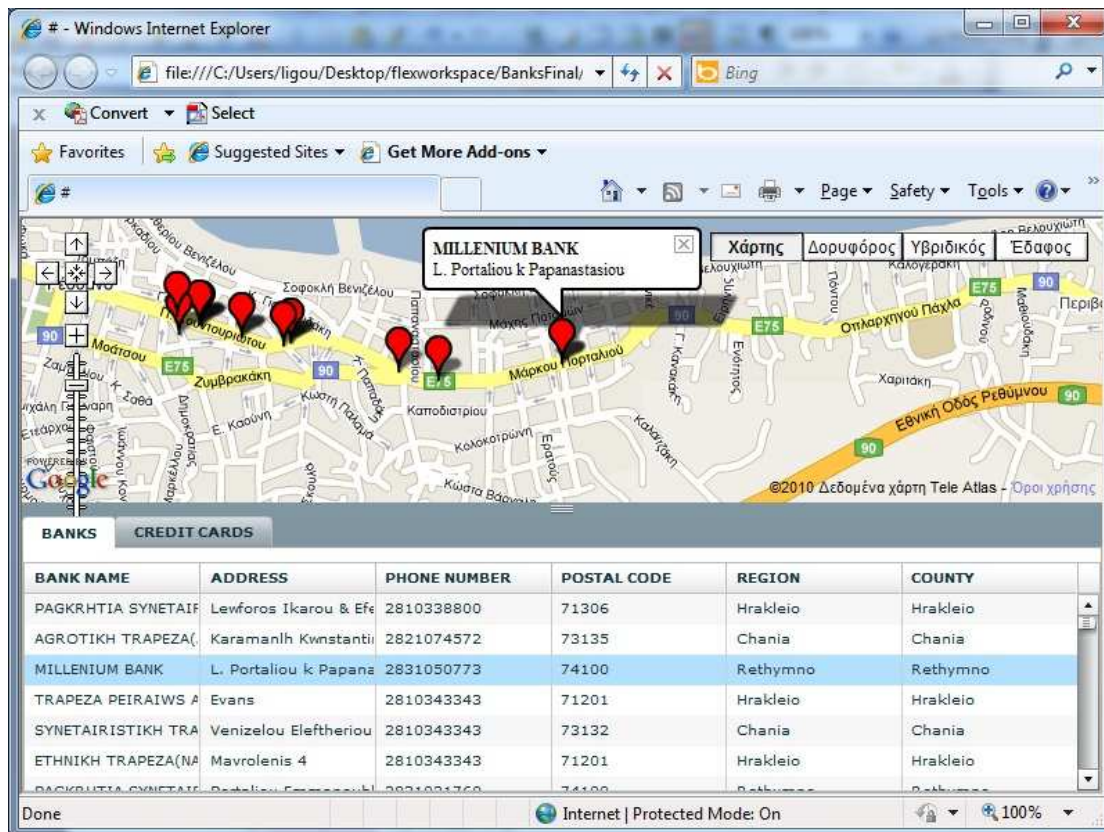
Η συνάρτηση `readXml` για όλα τα στοιχεία του αρχείου `markers.xml` δηλώνει σαν μεταβλητές τα `marker`, `name`, `address`, `type` και `latlng` που υπάρχουν σε κάθε στοιχείο. Στη συνέχεια ανακαλεί την συνάρτηση `createMarker` η οποία παίρνει ως ορίσματα τις μεταβλητές `latlng`, `name`, `address` και `type`.

Συνάρτηση `createMarker()`:

Η συνάρτηση createMarker δημιουργεί ένα marker για όλα τα στοιχεία του markers.xml αρχείου πάνω στο χάρτη. Ορίζει το fillstyle των markers και με ένα event listener on click κάθε φορά που ο χρήστης κάνει κλικ σε ένα marker πάνω στον χάρτη θα εμφανίζεται ένα InfoWindow με το όνομα της τράπεζας που βρίσκεται στο σημείο που δείχνει ο marker και την διεύθυνση της.

Συνάρτηση onGridItemClick():

Με αυτήν την συνάρτηση κάθε φορά που ο χρήστης διαλέξει μια τράπεζα από το grid στο κάτω μέρος της εφαρμογής, μεταφέρεται το κέντρο του χάρτη αυτόματα στο σημείο που βρίσκετε η τράπεζα.



Συνάρτηση toggleCategory():

Η συνάρτηση toggleCategory ρυθμίζει εάν θα είναι ορατά ή όχι τα markers πάνω στον χάρτη ανάλογα με το type που έχει ο κάθε marker (visa, maestro, dinners ή American).

Τα Interfaces :

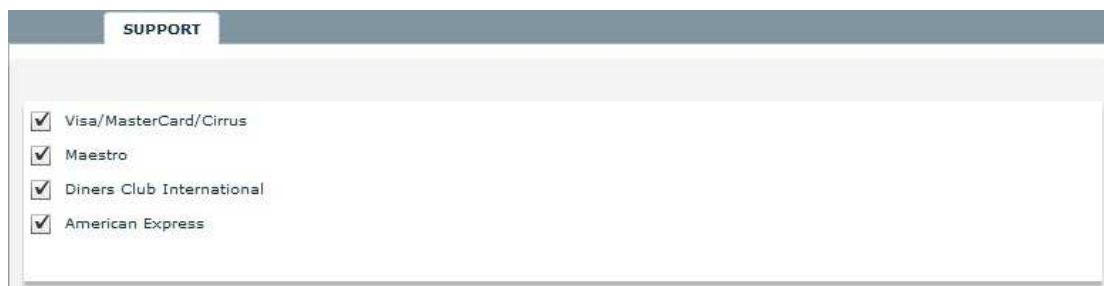
VDividedBox, Canvas, Panel, UIComponent, HDividedBox, VBox, Hbox και TabNavigator χρησιμοποιήθηκαν για να κάνουν εφικτή την διεπαφή του χρήστη με τις λειτουργίες της εφαρμογής.

Σε ένα VDividedBox προσθέσαμε ένα UIComponent στο πάνω μέρος στο οποίο θα εμπεριέχεται ο χάρτης. Με την εντολή resize="onContainerResize(event)" ο χάρτης θα προσαρμόζεται πάντα στο μέγεθος του container. Στο κάτω μέρος του VDividedBox προσθέσαμε ένα TabNavigator:



Στο TabNavigator προσθέσαμε 2 Canvas, το BANKS και το SUPPORT:

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
AGROTIKH TRAPEZA	Hg. Gavrihl 12	2831022267	74100	Rethymno	Rethymno
SYNETAIRISTIKH TR	L. Kountouriwτου 75	2831021030	74100	Rethymno	Rethymno
MILLENIUM BANK	L. Portaliou k Papana	2831050773	74100	Rethymno	Rethymno
ETHNIKH TRAPEZA(N	Palaja Ethnikh Odos I	2834022289	74052	Perama	Rethymno
ALPHA BANK	Lewforos Kountouriwt	2831035868	74100	Rethymno	Rethymno
ETHNIKH TRAPEZA(N	Kountouriwtου 84	2831055228	74100	Rethymno	Rethymno
ΠΑΚΙΟΥΤΑ ΣΥΝΕΤΑ	Παλιού Εμπορική	2831021760	74100	Rethymno	Rethymno



Όπως βλέπουμε στο tab BANKS έχουμε συμπεριλάβει το datagrid με τις πληροφορίες για τις τράπεζες ενώ στο tab SUPPORT ένα panel με Hbox και CheckBox.

Πηγή Δεδομένων Datagrid:

Σε αυτό το κομμάτι κώδικα ορίζουμε τον αριθμό των columns για το datagrid που θα δημιουργήσουμε, τις επικεφαλίδες τους οι οποίες είναι “BANK NAME”, “ADDRESS”, “PHONE NUMBER”, “POSTAL CODE”, “REGION” και “COUNTY” και τέλος ορίζουμε σε ποιο πεδίο του datagrid που έχουμε δημιουργήσει παραπάνω αντιστοιχεί στο κάθε column. Για παράδειγμα το column “BANK NAME” θα συμπληρωθεί από τα δεδομένα του datagrid που έχουν την ετικέτα location.

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
PAGKRHTIA SYNETAIF	Lewforos Ikarou & Efe	2810338800	71306	Hrakleio	Hrakleio
AGROTIKH TRAPEZA(C	Karamanlh Kwnstanti	2821074572	73135	Chania	Chania
MILLENIU BANK	L. Portaliou k Papan	2831050773	74100	Rethymno	Rethymno
TRAPEZA PEIRAIWS A	Evans	2810343343	71201	Hrakleio	Hrakleio
SYNETAIRISTIKH TRA	Venizelou Eleftheriou	2810343343	73132	Chania	Chania
ETHNIKH TRAPEZA(NA	Mavrolenis 4	2810343343	71201	Hrakleio	Hrakleio
PAGKRHTIA SYNETAIF	Portaliou Epistasiou	2831050773	74100	Rethymno	Rethymno

Αρχείο Markers.xml:

Το αρχείο Markers.xml είναι ένα ανεξάρτητο αρχείο .xml που έχουμε δημιουργήσει το οποίο περιέχει τις πληροφορίες και τα γεωγραφικά δεδομένα όλων των τραπεζών που θέλουμε να προσθέσουμε στην εφαρμογή. Το αρχείο markers.xml θα χρησιμοποιηθεί από τις συναρτήσεις readXml() και createMarker() έτσι ώστε να δημιουργηθούν τα markers πάνω στον χάρτη. Το markers.xml περιέχει πληροφορίες όπως: marker name στο οποίο βάζουμε το όνομα της τράπεζας, address στο οποίο βάζουμε την διεύθυνση της τράπεζας, το πεδίο pc που αντιστοιχεί στον ταχυδρομικό κώδικα της περιοχής που βρίσκεται η τράπεζα, phone το τηλέφωνο επικοινωνίας με την τράπεζα, τα region και county που βρίσκεται η τράπεζα, τις γεωγραφικές συντεταγμένες τις οποίες βρίσκεται η τράπεζα πάνω στον χάρτη και τέλος ένα πεδίο type που παίρνει τις τιμές visa, maestro, dinners και American και ορίζει τους τύπους που συνεργάζεται η κάθε τράπεζα.

```

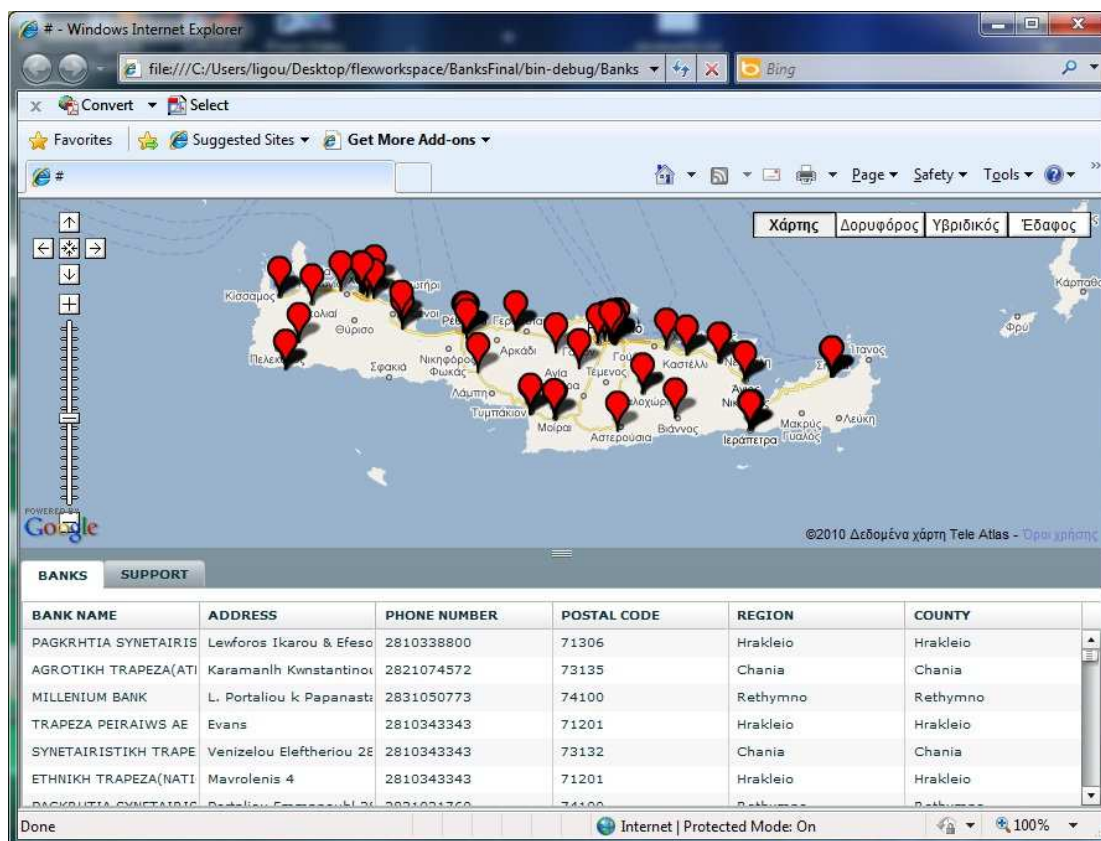
BanksFinal.mxml  markers.xml X
1<?xml version="1.0" encoding="UTF-8"?>
2<markers>
3<marker name="LAIKH TRAPEZA (ELLAS) AE" address="Idhs 26"
4phone="2810285300" pc="71201" region="Hrakleio" county="Hrakleio" lat="35.3389035" lng="25.1324448" type="visa"/>
5<marker name="PAGKRHTIA SYNETAIRISTIKH TRAPEZA(PANCRETA BANK)" address="Lewforos Ikarou & Efesus"
6phone="2810338800" pc="71306" region="Hrakleio" county="Hrakleio" lat="35.339074" lng="25.140476" type="visa"/>
7<marker name="AGROTIKH TRAPEZA (ATEBANK)" address="Karamanlh Kwnstantinou 16"
8phone="2821074572" pc="73135" region="Chania" county="Chania" lat="35.5027597" lng="24.027828" type="visa"/>
9<marker name="MILLENIU BANK" address="L. Portaliou k Papanastasiou"
10phone="2831050773" pc="74100" region="Rethymno" county="Rethymno" lat="35.364859" lng="24.4897037" type="visa"/>
11<marker name="MILLENIU BANK" address="L. Portaliou k Papanastasiou"
12phone="2831050773" pc="74100" region="Rethymno" county="Rethymno" lat="35.364859" lng="24.4897037" type="maestro"/>
13<marker name="TRAPEZA PEIRAIWS AE" address="Evans"
14phone="2810343343" pc="71201" region="Hrakleio" county="Hrakleio" lat="35.3364727" lng="25.1342057" type="visa"/>
15<marker name="ALPHA BANK" address="25hs Avgoustou 94"
16phone="2810229562" pc="71201" region="Hrakleio" county="Hrakleio" lat="35.3398155" lng="25.1338045" type="visa"/>
17<marker name="ALPHA BANK" address="25hs Avgoustou 94"
18phone="2810229562" pc="71201" region="Hrakleio" county="Hrakleio" lat="35.3398155" lng="25.1338045" type="american"/>
19<marker name="EUROBANK ERGASIAS AE" address="E. Venizelou 27"
20phone="2843023960" pc="72057" region="Shteia" county="Lasithiou" lat="35.2082902" lng="26.1064158" type="visa"/>
21<marker name="EUROBANK ERGASIAS AE" address="E. Venizelou 27"
22phone="2843023960" pc="72057" region="Shteia" county="Lasithiou" lat="35.2082902" lng="26.1064158" type="maestro"/>
23<marker name="ETHNIKH TRAPEZA (NATIONAL BANK)" address="Lewforos Ikarou 44 Neas Alikarnassou"
24phone="2810283456" pc="71601" region="Hrakleio" county="Hrakleio" lat="35.3387409" lng="25.1417652" type="visa"/>
25<marker name="ETHNIKH TRAPEZA (NATIONAL BANK)" address="Lewforos Ikarou 44 Neas Alikarnassou"
26phone="2810283456" pc="71601" region="Hrakleio" county="Hrakleio" lat="35.3387409" lng="25.1417652" type="maestro"/>
27<marker name="SYNETAIRISTIKH TRAPEZA XANIWN" address="L. Kountouriotou 75"
28phone="2831021030" pc="74100" region="Rethymno" county="Rethymno" lat="35.365809" lng="24.4785465" type="visa"/>

```


ΚΕΦΑΛΑΙΟ 6 Έλεγχος

6.1 Αναλυτική παρουσίαση ελέγχου

Αρχικά με την εκκίνηση της εφαρμογής μας εμφανίζεται το παρακάτω interface:



Στο πάνω αριστερό μέρος της εφαρμογής παρατηρούμε ότι έχει γίνει αυτόματα zoom στο νησί της Κρήτης. Ο χάρτης αποτελείται από controls τα οποία βοηθούν στην πλοήγηση και στο επιπλέον zoom. Στο πάνω δεξί μέρος της εφαρμογής παρατηρούμε το map type control το οποίο αποτελείται από 4 επιλογές τύπου προβολής χάρτη:

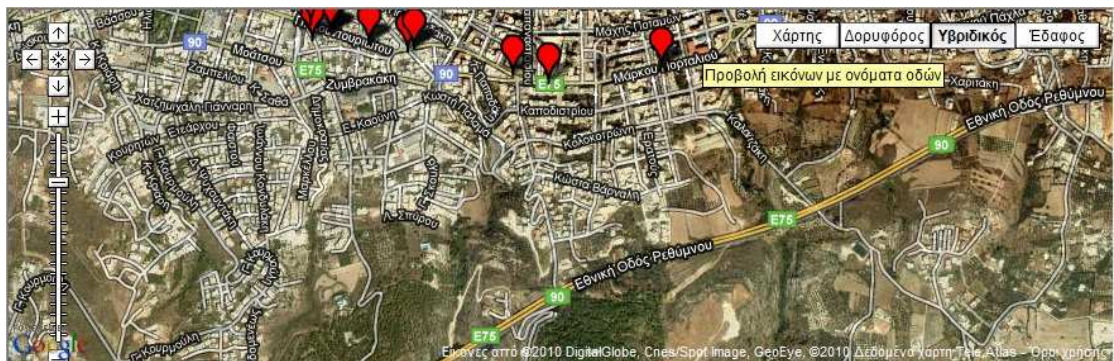
Χάρτης : Στην επιλογή χάρτης, στον χάρτη παρουσιάζονται σχηματικά οι οδοί, τα ονόματα των περιοχών, των οδών, των πόλεων, των κομοπόλεων και των χωριών καθώς και η χλιομέτρηση των οδών.



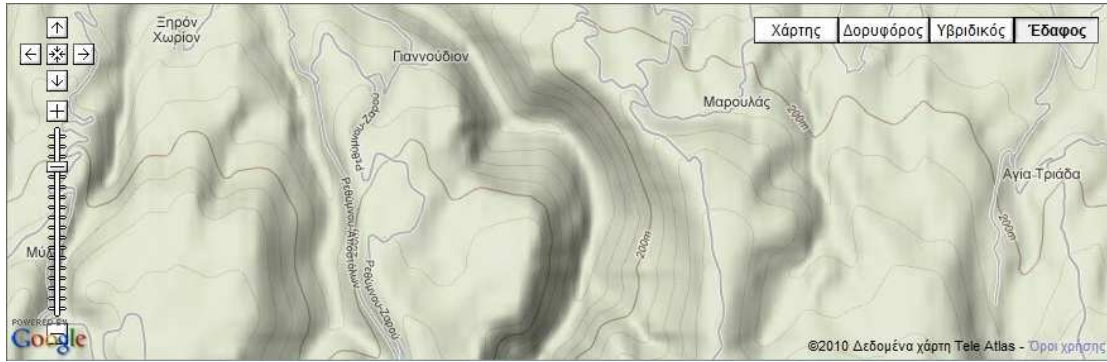
Δορυφόρος: Στην επιλογή δορυφόρος παρουσιάζεται ο χάρτης χωρίς ονόματα όπως ακριβώς φαίνεται από τον δορυφόρο.



Υβριδικός: Σε αυτήν την επιλογή έχουμε τον συνδυασμό των επιλογών Χάρτης και Δορυφόρος σε ένα χάρτη που περιέχει και τα δύο.



Έδαφος: Στην επιλογή έδαφος παρουσιάζεται ο χάρτης ανάλογα με τις υψομετρικές του ιδιότητες.



Κατά μήκος της Κρήτης έχουν τοποθετηθεί markers. Κάθε marker αντιστοιχεί στην ακριβή θέση μιας τράπεζας στην Κρήτη. Εάν ο χρήστης κάνει κλικ με το ποντίκι πάνω σε ένα marker τότε εμφανίζεται ένα InfoWindow με το όνομα και την διεύθυνση της τράπεζας της οποίας η θέση βρίσκεται στο σημείο που έχει τοποθετηθεί ο marker.



Στο κάτω μέρος της εφαρμογής υπάρχει ένα Tab Navigator με δύο επιλογές:

TAB BANKS: Στο tab BANKS υπάρχει ένα grid το οποίο περιέχει τις τράπεζες οι οποίες έχουν τοποθετηθεί με marker πάνω στον χάρτη και όλα τα στοιχεία τους (όνομα, διεύθυνση, τηλέφωνο επικοινωνίας, ταχυδρομικός κώδικας, πόλη, νομός).

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
PAGKRHTIA SYNETAIF	Lewforos Ikarou & Efe	2810338800	71306	Hrakleio	Hrakleio
AGROTIKH TRAPEZA	Karamanlh Kwnstanti	2821074572	73135	Chania	Chania
MILLENIUM BANK	L. Portaliou k Papana	2831050773	74100	Rethymno	Rethymno
TRAPEZA PEIRAIWS A	Evans	2810343343	71201	Hrakleio	Hrakleio
SYNETAIRISTIKH TRA	Venizelou Eleftheriou	2810343343	73132	Chania	Chania
ETHNIKH TRAPEZA(NA	Mavrolenis 4	2810343343	71201	Hrakleio	Hrakleio
PAGKRHTIA SYNETAIF	Rethymno Epimnidou	2831031366	74100	Rethymno	Rethymno

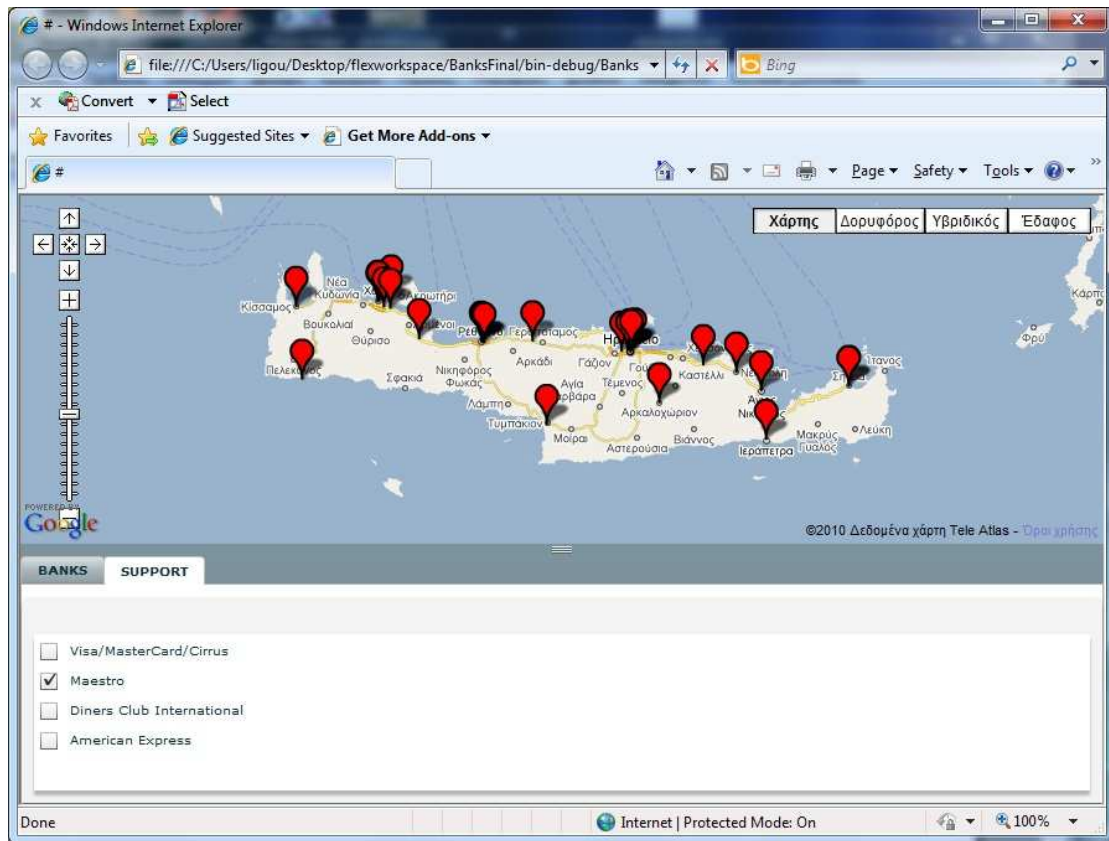
Εάν ο χρήστης κάνει κλικ με το ποντίκι πάνω στις πληροφορίες μιας τράπεζας πάνω στο grid τότε ο χάρτης μεταφέρει αυτόματα το κέντρο του στην τράπεζα που έχει επιλέξει ο χρήστης.

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
PAGKRHTIA SYNETAIRIS	Lewforos Ikarou & Efeso	2810338800	71306	Hrakleio	Hrakleio
AGROTIKH TRAPEZA(ATI)	Karamanli Kwnstantinou	2821074572	73135	Chania	Chania
MILLENIUM BANK	L. Portaliou k Papanastasi	2831050773	74100	Rethymno	Rethymno
TRAPEZA PEIRAIWS AE	Evans	2810343343	71201	Hrakleio	Hrakleio
SYNETAIRISTIKH TRAPE	Venizelou Eleftheriou 2E	2810343343	73132	Chania	Chania
ETHNIKH TRAPEZA(NATI)	Mavrolenis 4	2810343343	71201	Hrakleio	Hrakleio

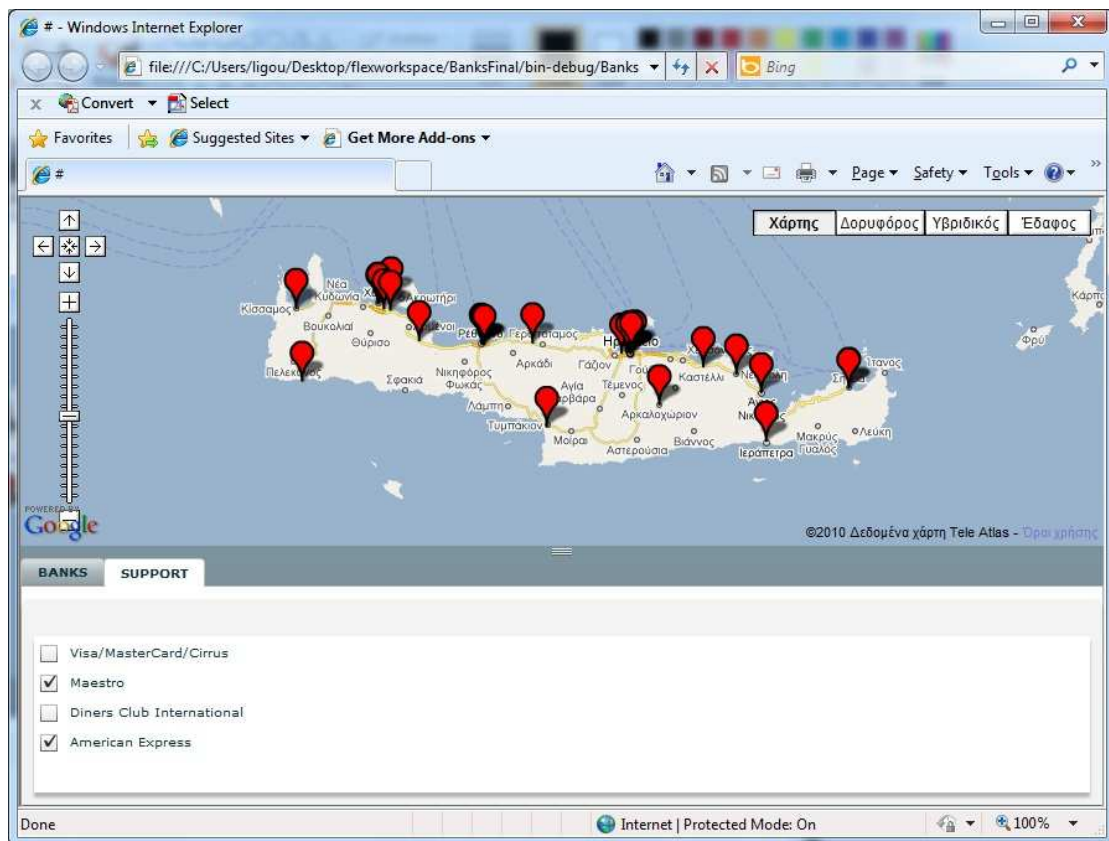
Ακόμα ο χρήστης έχει την δυνατότητα ταξινόμησης του grid ανάλογα με την συγκεκριμένη πληροφορία της τράπεζας που θέλει. Πχ ανά νομό:

BANK NAME	ADDRESS	PHONE NUMBER	POSTAL CODE	REGION	COUNTY
AGROTIKH TRAPEZA(ATI)	Karamanli Kwnstantinou	2821074572	73135	Chania	Chania
TRAPEZA KYPROU(CYPR)	Str. Tzanakakh 94	2821027844	73134	Chania	Chania
TRAPEZA KYPROU(CYPR)	Kydwvias 79	2821026270	73134	Chania	Chania
SYNETAIRISTIKH TRAPE	Venizelou Eleftheriou 2E	2810343343	73132	Chania	Chania
ETHNIKH TRAPEZA(NATI)	Lewforos El. Venizelou	2821064486	73100	Akrwthri Xaniwn	Chania
AGROTIKH TRAPEZA(ATI)	Str. Tzanakakh 82		73100	Chania	Chania

TAB SUPPORT: Στο tab SUPPORT υπάρχουν 4 combo box. Ο χρήστης έχει την δυνατότητα να επιλέξει το combo box που επιθυμεί για να εμφανίζονται στον χάρτη μόνο οι τράπεζες που συνεργάζονται με την επιλογή του. Πχ Maestro



Ακόμα ο χρήστης έχει την δυνατότητα να εμφανίσει στο χάρτη και δύο, τρεις ή και όλες τις κατηγορίες ταυτόχρονα.



ΚΕΦΑΛΑΙΟ 7 Υλοποίηση

7.1 Πλατφόρμες και προγραμματιστικά εργαλεία

7.1.1 Adobe Flash Platform

Η Adobe Flash πλατφόρμα είναι ένα ολοκληρωμένο σύνολο τεχνολογιών που περιβάλλεται από ένα καθιερωμένο σύστημα προγραμμάτων στήριξης, από επιχειρηματικές εταιρίες και από ενθουσιώδης κοινότητες χρηστών. Όλα αυτά μαζί, παρέχουν όλα όσα χρειάζεστε για να δημιουργήσετε και να παραδώσετε συναρπαστικές εφαρμογές, δεδομένα, καθώς και βίντεο στο ευρύτερο δυνατό κοινό.

7.1.2 Adobe Flex Builder version 3

Το Adobe Flex είναι ένα κιτ ανάπτυξης λογισμικού (SDK) που δημιουργήθηκε από την Adobe Systems για την ανάπτυξη και εγκατάσταση cross-platform πλούσιων εφαρμογών Διαδικτύου που βασίζονται στην πλατφόρμα Adobe Flash. Οι Flex εφαρμογές μπορεί να γραφτούν χρησιμοποιώντας το Adobe Flash Builder ή χρησιμοποιώντας τον ελεύθερα διαθέσιμο Flex compiler από την Adobe.

Η αρχική έκδοση το Μάρτιο του 2004 της Macromedia περιλαμβάνει ένα SDK, ένα IDE, και μια εφαρμογή Java EE integration γνωστή ως Flex Data Services. Δεδομένου ότι η Adobe αγόρασε την Macromedia, το 2005, οι επόμενες εκδόσεις του Flex δεν απαιτούν άδεια για το Flex Data Services, η οποία έχει αναδειχθεί σε ξεχωριστό προϊόν και μετονομάστηκε σε LiveCycle Data Services.

Το Φεβρουάριο του 2008, η Adobe κυκλοφορήσει το SDK Flex 3 στο πλαίσιο του ανοιχτού κώδικα Mozilla Public License και έτσι οι Flex εφαρμογές μπορούν να αναπτυχθούν χρησιμοποιώντας οποιοδήποτε πρότυπο IDE, για παράδειγμα το Eclipse. Υπάρχει επίσης μια εμπορική και βιομηχανική ιδιοκτησία IDE που ονομάζεται Adobe Flash Builder.

Παραδοσιακοί προγραμματιστές εφαρμογών βρήκαν δύσκολο να προσαρμοστούν στην φιλοσοφία του Animation στην οποία η Flash πλατφόρμα είχε αρχικά σχεδιαστεί. Το Flex επιδιώκει να ελαχιστοποιήσει το πρόβλημα αυτό με την παροχή μιας ροής εργασίας και ενός μοντέλο προγραμματισμού στα οποία είναι εξοικειωμένοι οι προγραμματιστές.

Το Flex SDK έρχεται με ένα σύνολο στοιχείων διεπαφής χρήστη όπως buttons, list boxes, trees, data grids, διάφορα text controls, και αρκετά layout containers. Τα Charts και τα graphs είναι διαθέσιμα σαν add-on.

7.1.3 MXML

Η MXML που είναι βασισμένη στην XML γλώσσα προγραμματισμού εισήχθη για πρώτη φορά από την Macromedia τον Μάρτιο 2004. προσφέρει έναν τρόπο να οικοδομήσουμε γραφικές διεπαφές χρήστη. Η Διαδραστικότητα επιτυγχάνεται με τη χρήση του ActionScript, την γλώσσα πυρήνα του Flash Player που είναι βασισμένο στο πρότυπο ECMAScript. Η Adobe Systems (η οποία απέκτησε την Macromedia, το Δεκέμβριο του 2005) δεν παρέχει κανένα επίσημο νόημα για το ακρωνύμιο, αλλά μερικοί προγραμματιστές κάνουν εικασίες όπως "Magic eXtensible Markup Language". Είναι πιθανό ότι το όνομα προέρχεται από το MX κατάληξη που έχει δοθεί σε προϊόντα όπως το Macromedia Studio που κυκλοφόρησε το 2002 και το 2004, ή απλά "Macromedia eXtensible Markup Language". Οι προγραμματιστές εφαρμογών χρησιμοποιούν την MXML σε συνδυασμό με την ActionScript για να αναπτύξουν πλούσιες διαδικτυακές εφαρμογές, με τη βοήθεια εφαρμογών όπως το Adobe Flex.

Η MXML μπορεί να περιέχει κομμάτια κώδικα ActionScript, είτε κατά τη δημιουργία του body ενός event handler function, ή με data binding εφόσον τα άγκιστρα () χρησιμοποιούνται στην σύνταξη.

7.2 Διαδικασία εγκατάστασης σε υπολογιστή

Για να γίνει η εκτέλεση της εφαρμογής δεν χρειάζεται ο χρήστης να κάνει εγκατάσταση στον υπολογιστή. Θα πρέπει όμως να έχει ήδη εγκατεστημένο το Adobe Flash Player το οποίο είναι διαθέσιμο στην ηλεκτρονική διεύθυνση <http://get.adobe.com/flashplayer/>. Θα πρέπει ακόμα όταν ο χρήστης τρέξει την εφαρμογή να υπάρχει στον ίδιο φάκελο το αρχείο markers.xml. Σε ορισμένα λειτουργικά συστήματα όπως στα Windows το Flash Player σαν πρόσθετη ασφάλεια δεν επιτρέπει την εκτέλεση εξωτερικών αρχείων όπως στην συγκεκριμένη περίπτωση του markers.xml οπότε θα πρέπει να το ορίσετε εσείς ως exeption αν θέλετε να τρέξετε το πρόγραμμα σαν αυτόνομη εφαρμογή:

Θα πρέπει να πλοηγηθείτε στο path:

Windows: System \ Macromed \ Flash \ FlashPlayerTrust

(Για παράδειγμα, C: \ Windows \ system32 \ Macromed \ Flash \ FlashPlayerTrust)

Mac: app support / Macromedia / FlashPlayerTrust

(Για παράδειγμα, το / Library / Application Support / Macromedia / FlashPlayerTrust)

Στην συνέχεια δημιουργήστε ένα αρχείο στο notepad και αποθηκεύστε το σε μορφή .cfg βάζοντας μέσα την διεύθυνση που έχετε αποθηκεύσει την εφαρμογή στον σκληρό δίσκο του υπολογιστή σας



Εάν συμπεριλάβετε την εφαρμογή σε μια ιστοσελίδα τότε δε χρειάζεται να ακολουθήσετε την παραπάνω διαδικασία.

ΚΕΦΑΛΑΙΟ 8 Βιβλιογραφία

C. Korhonen, D. Hassoun, J. Crosby. “Creating Mashups with Adobe Flex and AIR” 2008

“Google Maps API for Flash”

<http://code.google.com/apis/maps/documentation/flash/intro.html>

A. Cole “O'Reilly Learning Flex.3” 2008