



# **Xamarin Studio ja PhoneGapin sovel- tuvuus Rimeforgen käyttöön**

Mikko Piironen

Opinnäytetyö  
Syyskuu 2015  
Tietojenkäsittely  
Ohjelmistotuotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Ohjelmistotuotanto

PIIRONEN, MIKKO:

Xamarin Studion ja PhoneGapin soveltuvuus Rimeforgen käyttöön

Opinnäytetyö 66 sivua, joista liitteitä 38 sivua  
Syyskuu 2015

---

Tämän opinnäytetyön taustalla oli tamperelaisen pelialan yrityksen tarve saada tietoa PhoneGapista ja Xamarin Studiosta sekä laatia opas niiden käyttöönottoon. Työn tavoitteena oli selvittää PhoneGapin ja Xamarin Studion soveltuvuus Rimeforgen käyttöön ja tehdä opas PhoneGapin ja Xamarin Studion käyttöönotosta. Tutkimustehtävänä oli tehdä selvitys alustariippumattomasta ohjelmistotuotannosta. Selvitys tehtiin internetistä ja kirjallisuudesta löytyneen materiaalin ja käytännön tuoman tiedon avulla.

Tuloksena oli selvitys eri puhelinalustojen markkinaosuuksista maailmalla ja Suomessa sekä katsaus siihen, mitä on alustariippumaton ohjelmistotuotanto tällä hetkellä. Lisäksi koottiin tietoa PhoneGapin ja Xamarin Studion hyvistä ja huonoista puolista sekä laadittiin selvitys työkalujen käyttöönoton kannattavuudesta. Tuloksena oli myös käyttöönotto-opaste molemmista sovituisista työkaluista.

Johtopäätöksiksi tuli, että Xamarin on tällä hetkellä aloittelevalle yritykselle kallis sijoitus siitä saatavaan mahdolliseen hyötyyn nähden. PhoneGap on ilmainen, ja se tarjoaa riittävästi keinoja tehdä alustariippumattomia sovelluksia. Selvitetystä vaihtoehdoista PhoneGap tarjosi enemmän halvemmallä, ja suosittelenkin työn tilaajalle sen käyttöönottamista. Toisaalta ala on jatkuvassa muutoksessa. Tätä kirjoittaessa ilmestyivät esimerkiksi Windows 10 sekä Visual Studio 2015, ja nähtäväksi jää, kuinka nämä tuotteet muuttavat sitä, mihin Rimeforgen kannattaa suunnata tuotantoansa.

Kehittämisideoina on selvittää Visual Studio 2015:n ja Apache Cordovan käyttö PhoneGapin tilalla, tutustua Xamarin Studioon paremmin ja selvittää ne edellytykset, joilla Xamarin Studio kannattaa ottaa käyttöön Rimeforgella.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
School of Art, Music and Media  
Business Information Systems

**MIKKO PIIRONEN:**

Suitability of Xamarin Studio and PhoneGap for the Use of Rimeforge Entertainment

Bachelor's thesis 66 pages, appendices 38 pages  
September 2015

---

The objective of this study was to gather information about platform independent programming and to find out how PhoneGap and Xamarin would meet the needs of Rimeforge Entertainment as main platform independent programming tools for utility applications. The purpose was to create a manual for installing PhoneGap, Xamarin and additional software needed by these two programs to function properly.

This study was carried out as a project. In addition to existing knowledge gained through practical experience, all the information necessary for completing this study was gathered from internet, literature and knowledge gained by practical experience.

As a result, this study offers information about what platform independent programming is at the moment, and what the market shares of different platforms are in the world and in Finland. This study also offers review of pros and cons of PhoneGap and Xamarin Studio, and a manual of installation of both tools.

The findings indicate that Xamarin is a too expensive investment for Rimeforge in the current economic situation. PhoneGap is free of charge, and it provides enough tools to create platform independent applications. At the moment, Rimeforge should take PhoneGap as their primary platform independent programming tool.

While this study was being written, Microsoft published Windows 10 and Visual Studio 2015. It remains to be seen how these two programs will affect the situations as regards where Rimeforge should aim its programming resources.

As a development idea, Rimeforge should experiment to see whether it could use Visual Studio 2015 and Apache Cordova instead of PhoneGap. Rimeforge should also evaluate Xamarin Studio more widely and find out the conditions where starting to use Xamarin Studio would be cost-effective.

---

Key words: cross-platform, native application, hybrid application, PhoneGap, Xamarin Studio

## SISÄLLYS

1	JOHDANTO.....	4
2	OPINNÄYTETYÖN TAUSTA .....	5
2.1	Toimeksiantajan esittely .....	5
2.2	Tavoite ja tarkoitus .....	5
3	MOBIILIOHJELMOINTI.....	6
4	ALUSTARIIPPUMATON MOBIILIOHJELMISTOKEHITYS.....	9
5	TYÖKALUT .....	10
5.1	PhoneGap.....	10
5.1.1	Hyvät ja huonot puolet.....	13
5.1.2	Suorat kustannukset .....	14
5.1.3	Käyttöönotto.....	15
5.2	Xamarin Studio .....	17
5.2.1	Hyvät ja huonot puolet.....	18
5.2.2	Suorat kustannukset .....	18
5.2.3	Käyttöönotto.....	19
5.3	Ominaisuuksien vertailu .....	21
6	JOHTOPÄÄTÖKSET .....	23
7	POHDINTA.....	25
	LÄHTEET.....	26
	LIITTEET .....	28
	Liite 1. Sähköposti Xamarinilta.....	28
	Liite 2. Visual Studio -projektin muuttaminen PhoneGap projektiksi .....	29
	Liite 3. PhoneGap Käyttöönotto-ohjeistus .....	34
	Liite 4. Xamarin Käyttöönotto-ohjeistus.....	60

## LYHENTEET JA TERMIT

Alustariippumaton	Ohjelmointikieli tai sovellus, joka ei ole sidoksissa tiettyyn laitteistoalustaan tai käyttöjärjestelmään.
Apache Cordova	Avoimen lähdekoodin kehys mobiiliohjelmistojen kehittämiseksi. Kokoelma APIa, joilla pääsee käsiksi laitteistojen natiiveihin toimintoihin, kuten kameraan tai geolokaatioon JavaScriptillä.
API	Application Programming Interface eli ohjelmointirajapinta, määritelmä jonka mukaan eri ohjelmat voivat keskustella keskenään.
Back-end	Sovellus, joka toimii laitteiston ja front-end-sovelluksen välillä.
C#	Microsoftin kehittämä moderni olio-pohjainen ohjelmointikieli. Pohjautuu C-perheen kieliin, joten C:n, C++:n ja Javan käyttäjien on helppo oppia tämä kieli. C# on ECMA-334 ja ISO/IEC 23270 -standardien mukainen kieli.
CSS3	Cascading Style Sheet (porrastetut tyyliarkit) on WWW-dokumenteille kehitetty tyyliohjeiden laji, jota ylläpitää ja kehittää W3C-konsortio. CSS3 on CSS-tyylikielen uusin versio, joka on täysin yhteensopiva aiempien versioiden kanssa.
Front-end	Sovellus, jota käyttäjä käyttää ja joka on käyttäjän ja back-end-sovelluksen välissä.
HTML5	Hyper Text Markup Language 5 on verkkosivujen tekemiseen yleisesti käytetystä HTML-merkintäkielestä uusin versio. Siitä tuli W3C:n suositus 28.10.2014, jolloin se julkaistiin virallisesti.
Hybridi sovellus	Natiivikoodin ja HTML5-lähdekoodin yhdistelmä.
IDE	Integrated Development Environment eli Integroitu ohjelmointiympäristö, joita ovat esimerkiksi Visual Studio, Xamarin Studio tai Eclipse.
Javascript	Ohjelmointikieli, jota käytetään tavallisimmin selainpäässä lisäämään dynaamisuutta www-sivuille. Sitä käytetään usein

	myös palvelinten verkko-ohjelmoinnissa, pelien kehityksessä ja työpöytä- sekä mobiilisovellusten luomisessa.
Multi-tier	Useammasta kerroksesta koostuva sovellus, joita voidaan kehittää ja käyttää toisistaan riippumatta.
Natiivisovellus	Alustan omalla kehitysympäristöllä ja ohjelmointikielellä toteutettu sovellus.
PhoneGap	Ilmainen ja avoimen lähdekoodin viitekehys alustariippumattomien mobiilien ohjelmien ohjelmointiin. Adoben kehittämä Apache Cordovan jakeluversio.
Unity	3D- ja 2D pelien kehitysalusta.
User Experience UX	Käyttökokemus
User Interface UI	Käyttöliittymä
Visual Studio	Microsoftin kehittämä IDE sovelluskehitykseen.
Webview	Näyttää HTML-sisällön sovelluksessa.

## 1 JOHDANTO

Työn toimeksiantaja Rimeforge Entertainment on nuori tamperelainen peli- ja ohjelmistotalo, joka hakee alustariippumattomien hyötysovellusten ohjelmointiprojekteihin soveltuvaa työkalua. Markkinoilla on tarjolla runsaasti erilaisia työkaluja tällaisten sovellusten ohjelmointiin. Näitä työkaluja ovat muun muassa RhoMobile, PhoneGap, Appcelerator, MoSync ja WidgetPad.

Tavoitteena tällä työllä on selvittää, miten työssä tarkasteltaviksi valitut työkalut soveltuvat toimeksiantajan kehitysympäristöiksi ja tehdä käyttöönotto-opaste näille työkaluille. Tarkasteltavien työkalujen valintaan vaikutti vaatimus käytettävästä ohjelmointikielistä, hinta ja tuetut alustat.

Ala kehittyy jatkuvasti, ja näistä työkaluista kirjoitettu tieto saattaa vanhentua ohjelmistojen seuraavan päivituksen yhteydessä. Esimerkiksi tätä työtä kirjoitettaessa Windowsista on julkaistu versio 10 ja Visual Studiosta on julkaistu versio 2015. Nämä molemmat ohjelmistot voivat muuttaa painopistettä siinä, mitä työkaluja toimeksiantajan kannattaa ottaa käyttöön alustariippumattomien sovellusten ohjelmoinnissa.

Työhön on valittu tarkasteltaviksi työkaluiksi Xamarin Studio ja PhoneGap. Xamarin Studio on valittu tarkasteltavaksi työkaluksi siinä käytetyn ohjelmointikielen C#:n vuoksi. PhoneGap on valittu tarkasteltavaksi työkaluksi hintansa ja tukemiensa alustojen, siinä käytettävien tekniikoiden sekä toimeksiantajan työkalusta saaman aiemman mielikuvan perusteella. Muut mahdolliset työkalut rajattiin pois tarkasteltavien työkalujen joukosta yhteistyössä toimeksiantajan kanssa. Tässä työssä käydään läpi työkalujen asennuksen osalta vain Windowsille tehtävät asennukset puutteellisten resurssien vuoksi.

Aiheesta on saatavana paljon tietoa, mutta sitä ei ole löydy yhdestä paikasta, vaan aiheeseen tutustuminen vaatii paljon perehtymistä eri lähteisiin. Tämä työ pyrkii kokoamaan tietoa PhoneGapista ja Xamarin Studiosta luettavaksi yksiin kansiin aiheesta kiinnostuneiden saatavaksi.

## 2 OPINNÄYTETYÖN TAUSTA

### 2.1 Toimeksiantajan esittely

Rimeforge Entertainment on vuonna 2014 perustettu tamperelainen ohjelmisto- ja pelituotantoalan yritys. Rimeforgen tuotantoon kuuluvat 3D-mallinnukset ja animaatiot, web-palvelut, hyötysovellukset ja pelituotanto. Opinnäytetyön kirjoittamishetkellä Rimeforge työllistää neljä henkilöä.

Pelituotannossa Rimeforgessa käytetään tuotantoympäristönä Unityä ja Visual Studiota. Unity tukee ohjelmointikielinä C#:a ja UnityScriptiä (Unity Documentation 2015). Visual Studio 2013:ssa voi ohjelmointikielinä käyttää muun muassa Visual Basicia, C++:aa, C#:a tai HTML5/JavaScriptiä (Visual Studio 2015). Rimeforgessa ohjelmointikielenä Unityssä käytetään ensisijaisesti C#:a ja Visual Studiossa yksinomaan C#:a. Yrityksessä on kokemusta verkkoprojektien tiimoilta myös CSS:stä, HTML5:stä ja JavaScriptistä. Rimeforgen yrityskäyttöön tarkoitetut sovellukset ovat opinnäytetyön kirjoitusvaiheessa kotimaisille yrityksille suunniteltuja ja suunnattuja. Koska Windows Phonella on vielä tätä kirjoitettaessa vahva asema kotimaisilla mobiilimarkkinoilla, täytyy yrityksen huomioida ohjelmistojen kehitystyössä Androidin ja iOSin lisäksi Windows Phone.

Rimeforge hakee mobiilien hyötysovellusten tekemiseksi kustannustehokasta työkalua, jolla voidaan käyttää mahdollisimman paljon hyödyksi yrityksessä jo olevaa osaamista, ilman tarvetta uuden ohjelmointikielen opetteluun.

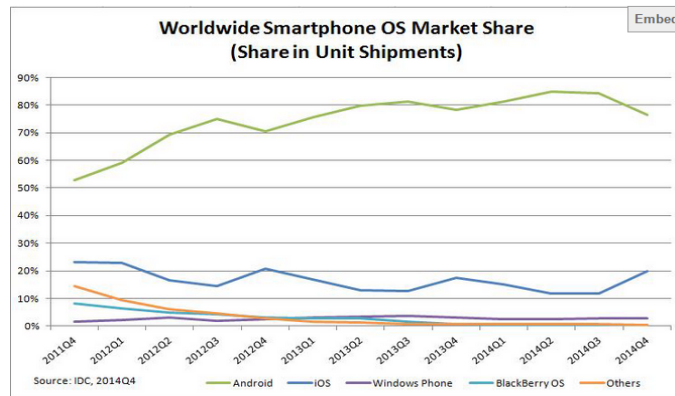
### 2.2 Tavoite ja tarkoitus

Tämän työn tavoitteena on selvittää Xamarin Studion sekä PhoneGapin soveltuvuus Rimeforgen käyttöön sekä arvioida, kuinka nämä työkalut kannattaa ottaa käyttöön Rimeforgella. Tämän työn tarkoituksena on saada lopputuotteiksi raportti Xamarin Studion ja PhoneGapin soveltuvuudesta Rimeforgen tuotantoympäristöön ja käyttöönotto-ohjeistus molemmista ohjelmistoista.



### 3 MOBILIOHJELMOINTI

Mobiiliohjelmoinnilla tarkoitetaan kaikille mobiileille laitteille suunnattujen sovellusten ohjelmointia. Mobiililaite on määritelmän mukaan kannettava, henkilökohtainen laite, joka kulkee mukana melkein koko ajan, jota on helppo ja nopea käyttää sekä jossa on jonkinlainen verkkoyhteys (Maximiliano Firtman, 2013). Älypuhelin on Oxfordin sanakirjan määritelmän mukaan matkapuhelin, joka suorittaa useita tietokoneen toimintoja, ja jossa on yleensä kosketusnäyttö, internet-yhteys ja käyttöjärjestelmä; näillä voidaan käyttää ladattuja sovelluksia (Smartphone 2015). Tässä opinnäytetyössä mobiiliohjelmoinnilla tarkoitetaan älypuhelimia varten suunniteltuja ja toteutettuja sovelluksia.



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q4 2014	76.6%	19.7%	2.8%	0.4%	0.5%
Q4 2013	78.2%	17.5%	3.0%	0.6%	0.8%
Q4 2012	70.4%	20.9%	2.6%	3.2%	2.9%
Q4 2011	52.8%	23.0%	1.5%	8.1%	14.6%

KUVA 1. Toimitetut puhelimet käyttöjärjestelmittäin 2011-2014 (Smartphone OS Market Share Q4 2014)

Markkinoilla on tällä hetkellä useita kilpailevia käyttöjärjestelmiä älypuhelimille. Näistä älypuhelinmarkkinoista suurimman markkinaosuuden vievät Android, iOS sekä Windows Phone -käyttöjärjestelmillä varustetut puhelimet. Kuvassa 1 kuvataan toimitettujen puhelinten lukumäärää käyttöjärjestelmittäin. IDC:n mukaan maailmanlaajuisesti laitetoimituksia käyttöjärjestelmittäin vertailtuna dominoi vuonna 2014 neljännellä kvartaalilla Android 76,6 % markkinaosuudella, toisena oli iOS 19,7 % markkinaosuudella ja kolmantena Windows Phone 2,8 % markkinaosuudella (Smartphone OS Market Share, Q4 2014). Saksassa Windows Phonen markkinaosuus oli jopa 8,2 % ja Italiassa 14,4 % (Smartphone OS sales market share 2015). Suomessa Androidin osuus markki-

noista vuonna 2014 oli 38 %, iOSin osuus oli 14 % ja Windows Phonen osuus Suomen markkinoista oli huomattavan korkea 32 % (Android Leads Finland's Smartphone Market (but Not by Much) 2015).

Android-sovellukset kirjoitetaan natiiveiksi sovelluksiksi käyttäen Javaa (Application Fundamentals 2015). iOS-sovellusten kirjoittamiseen käytetään Swiftiä, joka Applen mukaan perustuu C- ja Objective-C-ohjelmointikielien parhaisiin puoliin (About Swift 2015). Ennen Swiftin käyttöönottoa Objective-C oli pääasiallinen ohjelmointikieli iOS:lle, ja sitä voidaan edelleen käyttää sovellusten kirjoittamiseen (About Objective-C 2015). Windows Phone -sovellukset kirjoitetaan C#:lla, Visual Basicilla, Visual C++- ja JavaScript-kielillä tai näiden yhdistelmillä (Languages, tools and frameworks 2015).

Mikäli sovellukselle haluaa mahdollisimman suuren potentiaalisen asiakaskunnan, yrityksen kannattaa suunnata sovellus useammalle alustalle. Maailman mittakaavassa tämä tarkoittaa iOSia ja Androidia; Euroopan ja varsinkin Suomen mittakaavassa lisäksi Windows Phonea. Tällöin ongelmaksi muodostuvat sovelluksen tuotannossa tarvittavien henkilöresurssien määrä ja sovelluksen versioiden pitäminen samanlaisena eri alustoilla.

Tätä ongelmaa koetetaan ratkoa käyttämällä erilaisia työkaluja, jotka kääntävät sovelluksen kaikille alustoille joko natiiviksi tai hybridiksi sovellukseksi, tai sovellus kirjoitetaan suoraan HTML5-pohjaiseksi web-sivustoksi. Kaikilla näillä toteutustavoilla on omat hyvät ja huonot puolensa.

Natiivisovellus on sovellus, joka on kehitetty jotain tiettyä alustaa tai laitetta varten käyttäen alustan omaa kehitysympäristöä ja ohjelmointikieltä. Natiivisovellus toimii ainoastaan käyttöjärjestelmässä, jota varten se on tehty. Tällaisella sovelluksella on käytävissä kaikki alustan tarjoamat ominaisuudet kuten monikosketus, NFC, GPS ja kamera. Tällainen sovellus on kallis kehittää, mikäli tuki tarvitaan usealle alustalle. Käyttöliittymä on viiveettömin ja sujuvin, eikä se lähtökohtaisesti tarvitse yhteyttä verkkoon. (Mobiiliapplikaatioiden kehitys: HTML5, natiivi vai hybridi? 2015; Natiivi, hybridi ja HTML5 2015.) Sovelluksen UX on käyttäjälle helpoin sisäistää sovelluksen ollessa suoraan alustan vaatimusten mukaisesti alustaa varten ohjelmoitu.

HTML5-sovellus mobiililaitteille on sovellus, joka on optimoitu mobiililaitteille. Tällainen sovellus tavoittaa käytännössä kaikki älypuhelimien käyttäjät, ja sen kehittäminen on kustannustehokkaampaa kuin jokaiselle alustalle erikseen kehitetyt natiivit sovellukset. Sovellus voidaan tehdä vanhan sivuston tai verkkosovelluksen päälle, sille tehdyt päivitykset tulevat välittömästi käyttöön, ja niiden tekeminen on helppoa. Sovellus toimii selaimella, joten sitä ei tarvitse asentaa puhelimeen erikseen. Käyttöliittymän suhteen joudutaan tekemään kompromisseja, eivätkä kaikki alustan ominaisuudet ole käytössä. Sovelluksen ulkoasu ei ole natiivin sovelluksen näköinen, eikä sitä voida asentaa puhelimen muistiin applikaationa. Toimiakseen sovellus tarvitsee internet-yhteyden. (Mobiiliapplikaatioiden kehitys: HTML5, natiivi vai hybridi? 2015; Natiivi, hybridi ja HTML5 2015.)

Hybridisovellus on sovellus, jonka luomisessa on käytetty sekä käyttöjärjestelmän natiivia koodia, JavaScriptiä, että HTML5-pohjaista tekniikkaa. Sovelluksen sisälle luodaan nk. webview eli oma selain, jossa HTML5-sisältö näytetään. Hybridisovellus on edullinen kehittää, se tukee suurinta osaa puhelimen ominaisuuksista ja sitä voidaan jakaa keskitetysti käyttöjärjestelmän markkinapaikan kautta. Sovelluksen ulkoasu ei välttämättä vastaa natiivia sovellusta, ja vanhoilla laitteilla suorituskyky saattaa olla puutteellinen. (Mobiiliapplikaatioiden kehitys: HTML5, natiivi vai hybridi? 2015; Natiivi, hybridi ja HTML5 2015.)

## 4 ALUSTARIIPPUMATON MOBIILIOHJELMISTOKEHITYS

Useammalle alustalle natiivilla ohjelmointikielellä suunnatun ohjelmiston kehitys on hidasta ja kallista työtä, sillä koodi joudutaan tekemään jokaiselle alustalle erikseen. Tätä selvää puutetta kierretään kehittämällä ja käyttämällä työkaluja, joiden avulla voidaan kääntää tehty sovellus jokaiselle alustalle sopivaksi natiiviksi sovellukseksi, sovelluksesta tehdään hybridisovellus tai Web-pohjainen sovellus.

Alustariippumattomassa mobiiliohjelmistokehityksessä, niin kuin myös kaikessa muussa ohjelmistokehityksessä, on hyötyjä ja haittoja. Hyötyinä tällaisessa kehityksessä voidaan pitää esimerkiksi koodin uudelleenkäytettävyyttä.

Puhelimen ominaisuuksien hyödyntäminen on helppoa laajennuksien avulla, jos kehitystyökaluna käyttää esimerkiksi PhoneGapia tai Appceleratoria. Sovellusten kehittäminen on helppoa, koska jokaiselle alustalle voi kirjoittaa kielellä, jonka sovelluskehittäjä osaa jo valmiiksi, esimerkiksi Xamarinilla kielenä käytetään C#:a ja PhoneGapilla JavaScriptiä.

Haittoina alustariippumattomassa ohjelmoinnissa voidaan pitää muun muassa grafiikan ja 3D-ominaisuuksien rajallisuutta. Kaikkia alustojen ominaisuuksia ei välttämättä saada käyttöön, ja käytettävissä olevien työkalujen kehyksiä ei alustan toimesta tueta. Sovellus voi toimia huomattavasti hitaammin kuin natiivilla koodilla kirjoitettu sovellus, ja useimmat käytettävissä olevat työkalut käyttävät omia JavaScript-kirjastoja, jolloin työkalun vaihto voi olla suuren työn takana. (Cross-platform Mobile App Development - Good or Bad 2015).

Alustariippumatonta mobiiliohjelmointikehitystä varten on olemassa useita eri työkaluja. Tässä opinnäytetyössä keskitytään selvittämään työn tilaajan toiveiden mukaisesti PhoneGapin ja Xamarinin soveltuvuutta tilaajan käyttöön.

## 5 TYÖKALUT

### 5.1 PhoneGap

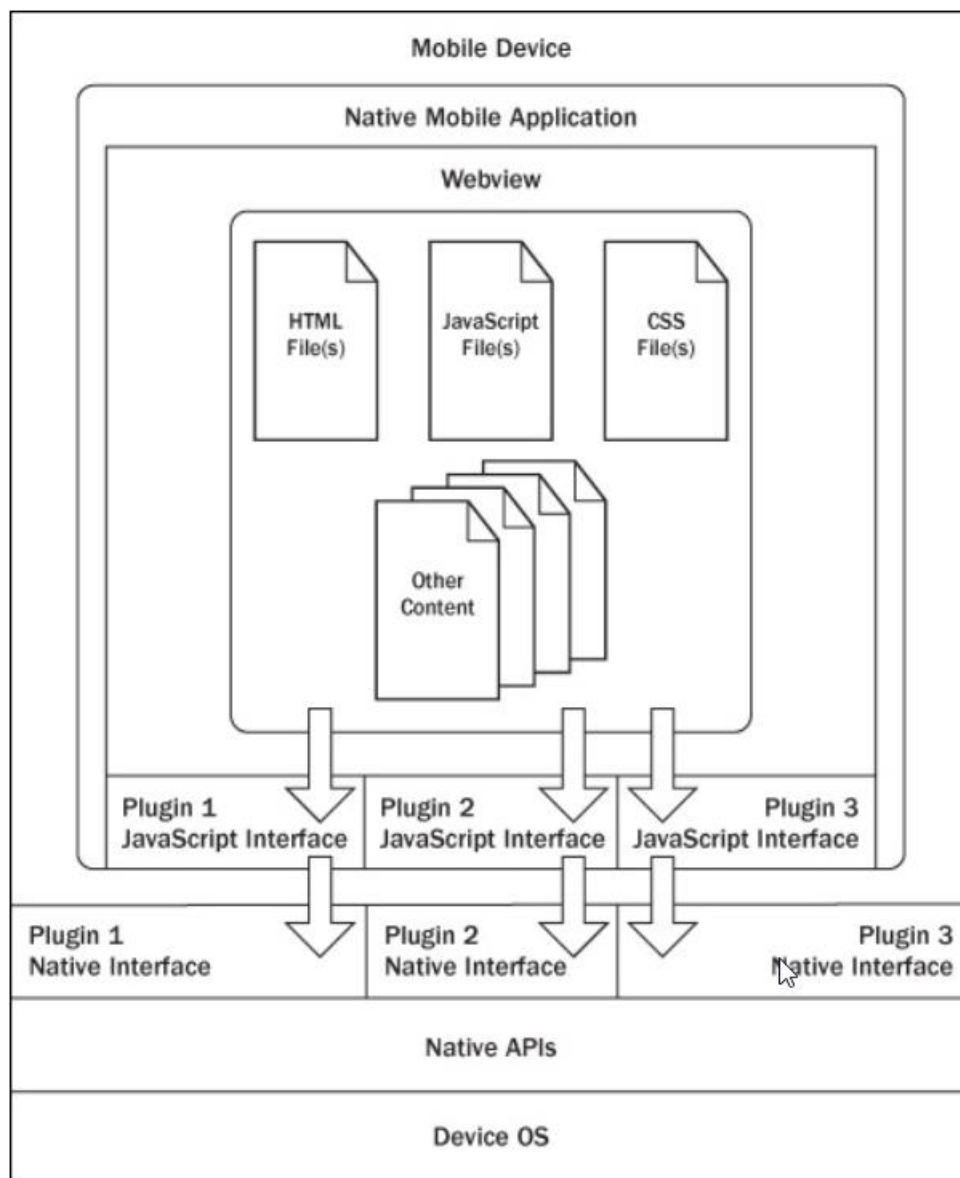
Apache Cordova on ilmainen, kenen tahansa käytettävissä oleva avoimen lähdekoodin kehys mobiiliohjelmistojen kehittämiseksi. PhoneGap on Adoben käyttämä jakeluversio Apache Cordovasta, ja ne ovat keskenään hyvin samankaltaisia. Kirjallisuudessa, kuten esimerkiksi Kerri Shottsin (2014) PhoneGap for Enterprise -kirjassa, näihin molempiin versioihin viitataan käyttämällä termiä Apache Cordova. Tässä työssä käytän tätä samaa lähestymistapaa, ja mikäli näiden versioiden välillä on eroavaisuuksia, tarkastelen asiaa PhoneGapin kannalta.

PhoneGap on ilmainen avoimen lähdekoodin työkalu ja se pohjautuu Apachen lisenssin versioon 2.0. Käyttäjä voi PhoneGapin avulla luoda CSS3:a, HTML5:tä ja JavaScriptiä käyttämällä mobiileja sovelluksia, käyttää puhelimen omia APIa ja kääntää ne useammalle mobiilille alustalle. Näitä alustoja ovat iPhone, Android, Blackberry, Windows Phone 8, Ubuntu ja Firefox OS. (PhoneGap, Cordova and what's in a name? 2015.)

Apache Cordova koostuu kolmesta komponentista. Ensimmäinen komponentti on natiivin sovelluksen lähdekoodin säiliö jokaiselle tuetulle mobiililaitteelle. Tämä säiliö muuntaa Cordovalla kirjoitetun sovelluksen laitteen ymmärtämään muotoon. Toinen komponentti on API-kokoelma; sen avulla web-sovellus pääsee käsiksi laitteen natiiveihin ominaisuuksiin, joihin ei normaalisti pääse käsiksi verkkoselaimella. Kolmas komponentti on kokoelma työkaluja, joilla hallitaan sovelluksen luontiprosessia, laajennosten elinkaarta, natiivien sovellusten kehittämistä ja testisovelluksia mobiililaitteiden simulaattoreihin ja emulaattoreihin. (Wargo, John M. 2014).

Cordova-sovellus luodaan tekemällä verkkosovellus, joka pakataan natiiviin säiliöön. Tämä sovellus testataan, siitä etsitään ja poistetaan virheet sekä se toimitetaan käyttäjille. Sovelluskehitys Cordovalla ei vaadi mitään erityisiä työkaluja, vaan sovelluksia voi tehdä tekstieditorilla kuten Notepadilla. (Wargo, John M. 2014).

John M. Wargon mukaan natiivissa Cordova-sovelluksessa sovelluksen käyttöliittymä (UI) koostuu yhdestä ruudusta, joka sisältää yksittäisen web-näkymän, joka käyttää kaiken käytettävissä olevan tilan laitteen näytöltä. Kuvassa 2 on kuvattu tällaisen sovelluksen arkkitehtuuri. Tyypillisesti mobiililla web-sovelluksella ei ole pääsyä laitteen sovelluksiin, laitteistoon tai natiiveihin APIhin. Cordova tarjoaa sovelluskehittäjille käyttöön JavaScript-kirjastoa, jolla pääsee käyttämään laitteen ominaisuuksia web-näkymän ulkopuolelta. Tämä JavaScript-kirjasto toimii yhdistämällä sovelluksen käyttämän koodin ja natiivin koodin, jolla laitteen API:t toimivat. Tämä kaikki toteutetaan yhdellä JavaScript-kirjastolla, jossa on erilliset natiivit toteutukset jokaiselle tuetulle mobiilille alustalle. (Wargo, John M. 2014).



KUVA 2. Apache Cordova 3.0. Sovelluksen arkkitehtuuri (Wargo, John M. 2014)

Apache Cordovassa APIt on jaettu erillisiksi laajennuksiksi, jotka voidaan tarvittaessa lisätä sovellukseen tai poistaa siitä. PhoneGap tukee suoraan ohessa olevan kuvan mukaisesti käyttöjärjestelmien APIa (kuva 3). Näitä ovat mm. kiihtyvyyssanturit, kamera ja geolokaatio.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	Windows Phone 8	Ubuntu	Firefox OS
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	✓	✓	X
Network	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓

✓ - supported feature  
X - unsupported feature due to hardware or software restrictions

KUVA 3. Adobe PhoneGap, tuetut APIt (Supported Features 2015)

PhoneGap- ja Visual Studio 2013 Apache Cordova -projektien eroina on sovelluksen määrittely sovelluksen konfigurointitiedostossa config.xml. Jotta Visual Studiolla tehty projekti voidaan muuttaa PhoneGap-projektiksi, täytyy msdn:n sivuston mukaan tehdä viisi asiaa. Näitä ovat uuden Cordova-projektin luominen Visual Studiolla, PhoneGap-projektin rakenteen luominen, config.xml-tiedoston päivittäminen, sovelluksen lataaminen pilvipalveluun ja koodin allekirjoittaminen sekä sovelluksen jakelu. Sama pätee myös Visual Studio 2015 RC:llä tehtyihin sovelluksiin. Näistä tehtävistä vaiheista löytyy ohjeistus liitteestä 2.

### 5.1.1 Hyvät ja huonot puolet

Kerri Shottsin (2014) mukaan yrityksen kannattaa valita PhoneGap useammasta syystä, joista hän listaa muutamia. Esille tulee mahdollisuus luoda nopeasti alustariippumaton hybridi sovellus pienellä päällekkäisellä työllä ja suurella määrällä koodin uudelleen käytettävyyttä. PhoneGapissa käytetään teknologioita, jotka ovat sellaisille kehittäjille tuttuja, jotka kehittävät web-, backend- ja multi-tier-sovelluksia. PhoneGapilla on pääsy laitteiden natiiveihin toimintoihin kuten verkkoon, sijaintiin ja kameraan. PhoneGapia varten voidaan luoda omia laajennuksia omia erityisiä tarpeita varten, ja sen voi sulauttaa osaksi natiivia sovellusta komponenttina. Kehitetyt sovellukset voidaan jakaa sisäisesti tai kauppapaikkojen kautta. PhoneGapilla on aktiivinen kehittäjäyhteisö, ja se on Apache-lisenssin alainen avoimen lähdekoodin kehys. Tarjolla on yritystason tukipalveluita, ja lisäksi Adobe tarjoaa yrityksille lisäominaisuuksia PhoneGapille osoitteessa <http://enterprise.phonegap.com>. (Shotts, Kerri, 2014).

Kerri Shottsin kirjassaan listaamien syiden lisäksi PhoneGapin etuina voidaan pitää sitä, että PhoneGapille on tehty valmiiksi paljon laajennuksia ja moduuleja, jotka tarjoavat ylimääräisiä palveluja tai työkaluja ilman tarvetta alkaa itse ohjelmoimaan niitä. Näistä laajennuksista löytyy kattava lista osoitteesta <https://build.phonegap.com/plugins>. PhoneGap on julkaissut PhoneGap Developer App -nimisen sovelluksen, jolla voi testata omia sovelluksia oikeissa laitteissa ilman koodin kääntämistä. Tämä toimii siinä tapauksessa, että sovelluksessa ei käytetä kolmansien osapuolien laajennuksia.

PhoneGap ei vaadi Rimeforgessa uuden ohjelmointikielen opettelua, siellä on jo nyt osaamista HTML5:stä, CSS3:sta ja JavaScriptistä. PhoneGap on ilmainen avoimen lähdekoodin työkalu, mikä on varsinkin aloittelevan yrityksen budjetissa iso tekijä.

PhoneGapilla tehty BMI-sovellus osoitti, että sama koodi toimii kokeilluilla Android- ja Windows Phone 8.1 -alustoilla ilman muutoksia. Toinen sovellus, jossa asennettiin geolokaatio-laajennus, tarkastettiin ja näytettiin laitteen sijainti, toimi myös moitteetta kokeilluilla alustoilla ilman tarvetta tehdä muutoksia koodiin, eikä kokeiltu sovellus tuntunut hitaammalta kuin vastaavat testatut natiivit sovellukset.



PhoneGapin huonoina puolina voidaan pitää muun muassa sitä, että sovelluksesta voi tulla hitaampi kuin natiivilla koodilla kirjoitetusta sovelluksesta. Tämä uhka toteutuu varsinkin sovelluksissa, joissa on paljon grafiikkaa. Koska laajennuksista suurin osa on tehty vapaaehtoisvoimin, voi oikein toimivien, viimeisimpien laajennusten löytäminen olla haastavaa. Kaikki tarjolla olevat laajennukset eivät toimi kaikilla alustoilla, jolloin näitä voi joutua itse ohjelmoimaan, ja tällöin tarvitaan alustakohtaisen ohjelmointikielen tuntemusta. PhoneGapin ilmaisessa versiossa sovelluksen suurin koko on 50 MB ja halvemmassa maksullisessa versiossa 100 MB. Tämä luonnollisesti on rajoittava tekijä suurempia grafiikkaa sisältäviä sovelluksia suunniteltaessa ja ohjelmoitaessa.

Molemmat PhoneGapilla tehdyt sovellukset osoittivat, ettei sovelluksesta tule käyttäjäkokemukseltaan natiivin oloista. Sovellukset osoittivat myös, etteivät niiden käyttöliittymät ole natiivin oloisia ja niiden muokkaaminen natiivin oloiseksi osoittautui liian haastavaksi tehtäväksi.

### **5.1.2 Suorat kustannukset**

PhoneGap on ilmainen sovellus. Ilmaisella versiolla ohjelmoija voi kehittää yhtä yksityistä sovellusta, jonka lähdekoodia muut eivät näe, ja rajattoman määrän avoimen lähdekoodin sovelluksia, jonka koodiin muut käyttäjät pääsevät tutustumaan esimerkiksi GitHubin kautta. Tällaisen sovelluksen koko voi enimmillään olla 50 MB.

PhoneGapista voi myös valita 9,99 dollaria kuukaudessa maksavan version. Tällöin käyttäjä saa kehittää 25 yksityistä sovellusta yhtä aikaa ja rajattoman määrän avoimen lähdekoodin sovelluksia. Sovelluksen suurin mahdollinen koko on tällöin 100 MB.

Jos käyttäjä liittyy Adoben pilvipalveluun ja maksaa vähintään 29,99 dollaria kuukaudessa, voi hän tehdä enintään 1 GB kokoisia sovelluksia. Kuvassa 4 on esitetty erilaiset hintavaihtoehdot PhoneGapille ja mitä kuhunkin hintaan sisältyy.

	Free Plan	Paid Plan	Adobe Creative Cloud Membership <a href="#">?</a>
open source apps	∞ unlimited must be pulled from a public Github repo		
private apps <a href="#">?</a>	1	25	
max app size	50 MB	100 MB	1 GB
core cordova plugins <small>* a list of these plugins is <a href="#">here</a></small>	YES		
third party plugins <small>* includes plugins from <a href="#">plugins.cordova.io</a> as well as our own repository</small>	YES		
upload plugins <small>* includes plugins only you can use</small>	NO	YES	
collaborators	∞ unlimited invite people to your app as either developers or testers		

KUVA 4. Adobe PhoneGap, hinnat (Choose your plan 2015)

### 5.1.3 Käyttöönotto

PhoneGapia voi käyttää joko komentorivikehotteesta CLI tai betatestaus-vaiheessa olevalla työpöytäsovelluksella. Jotta CLI voidaan asentaa, täytyy koneella olla asennettuna Node.js ja git client. CLIn avulla luodaan PhoneGap-projekti, siihen lisätään sovelluksessa käytettäviä laajennuksia, sekä sovellus käännetään eri alustoille. Sovellusten kääntäminen Android-alustalle vaatii Android-kehitystyökalujen asentamisen.

#### NodeJS:n asennus

Haetaan osoitteesta [nodejs.org](http://nodejs.org) NodeJS:n asennuspaketti ja käynnistetään ko. ohjelma. Hyväksytään License Agreement ja valitaan hakemisto, johon ohjelma halutaan asentaa. Varmistetaan, että asennusohjelma lisää NodeJS:n PATH-ympäristömuuttujiin ja asennetaan ohjelma. Asennus viimeistellään painamalla Finish-painiketta.

### Git clientin asennus

Ladataan ja asennetaan git client osoitteesta [git-scm.com](http://git-scm.com). Vastataan asennusohjelman esittämiin kysymyksiin ja käynnistetään ohjelman asennus. Lisätään ympäristömuuttujiin polku `Git\cmd-hakemistoon`.

### Android SDK:n asennus

Ladataan ja asennetaan osoitteesta [developer.android.com/sdk/index.html](http://developer.android.com/sdk/index.html) SDK-työkalut. Vastataan ruudulla näkyviin kysymyksiin ja aloitetaan asennus. Kun SDK-työkalut on asennettu, avataan ohjelma SDK Manager.exe ja asennetaan halutut Android API:t. Lisätään ympäristömuuttujiin polku Android-SDK:n `platform-tools-` ja `tools-hakemistoihin`.

### Windows Phone SDK:n asennus

Ladataan ja asennetaan Visual Studio 2015 tai muu Visual Studio alkaen VS 2013 -paketista. Tämä ohjelmisto sisältää Windows Phone SDK:n.

### PhoneGapin asennus

Avataan komentorivikehote ja kirjoitetaan komento `npm install -g phonegap`. Tällä komennolla tietokone hakee verkkoyhteyden kautta tarvittavat tiedostot ja asentaa PhoneGapin komentoriviversion tietokoneelle. PhoneGapin työpöytäversion voi asentaa lataamalla osoitteesta <http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/windows-asennusohjelmisto>. Lataamisen jälkeen käynnistetään ohjelma ja vastataan ruudulla esitettyihin kysymyksiin. Asennus viimeistellään painamalla Finish-näppäintä.

### PhoneGapin mobiiliohjelmiston asennus

Ladataan halutun puhelimen markkinapaikalta PhoneGap Developer -sovellus ja asennetaan se puhelimelle ominaisella tavalla. Tätä sovellusta, sen ollessa samassa verkossa tietokoneen kanssa, voidaan käyttää testaamaan reaaliajassa PhoneGapilla tehtyjä sovelluksia ilman tarvetta kääntää sovellusta tai asentaa laajennoksia. Tarkemmat ohjeet PhoneGapin käyttöön otosta on opinnäytetyön liitteessä 3.

## 5.2 Xamarin Studio

Xamarin Studio on IDE, jolla voi luoda sovelluksia iOSille, Macille ja Androidille. Xamarin Studio on integroitavissa Visual Studioon, mikäli ostaa business-lisenssin, jolloin samalla työkalulla voidaan kehittää sovelluksia myös Windows Phonelle. Vaikka Windows Phonelle kääntäminen vaatii Windows-tietokoneen ja iOS vaatii Mac OS X -tietokoneen, voidaan kaikille alustoille tehdä ohjelmointia samalla koneella. Sovellukset käännetään Windows Phonelle Macilla ohjelmoitaessa virtuaalisella Windows-koneella. Windowsilla ohjelmoitaessa sovellukset käännetään IPhonelle Macilla, joka on verkon kautta yhteydessä Windows-koneeseen.

Xamarin käyttää MVVM-mallia eli Model-View-ViewModel-mallia. Sovelluksen Model-kerros sisältää business-logiikan, jonka päällä sovellus toimii. View-kerros on käyttöliittymäkerros. Tämä kerros sisältää alustariippuvaisen koodin, jolla sovellusta käytetään. ViewModel-kerros välittää Model-kerrokselle View-kerroksen komennot ja Model-kerroksen antamat vastineet takaisin View-kerrokselle. (Piotr, J.Kula, 2014.)

Käytettäessä Xamarin.Forms:ia luodaan projekti, joka sisältää sovellusta varten yleisen kansion ja alustakohtaiset kansiot. Yleiseen kansioon tulee sovellusta varten jaettu koodi, ja alustakohtaiseen kansioon tulee alustakohtainen koodi. Sovelluksessa käytettävä jaettu koodi otetaan käyttöön luomalla alustakohtainen koodi, jolla laukaistaan jaettu koodi. Jokaisen alustan täytyy käyttää Xamarin.Forms-kehystä ja luoda instanssi ContentPagesta sovelluksen käynnistyessä. Esimerkiksi Androidissa luodaan Androidille tyypillinen Activity MainLauncher-attribuutilla, joka käynnistää jaetulla koodilla tehdyn sovelluksen.

Xamarinin verkkosivut lupaavat, että heidän viitekehystään käyttämällä sovelluksista saa luotua jopa 75 % käyttäen yhteistä koodia. Tätä väitettä ei voitu todentaa, sillä se olisi vaatinut suuremman projektin ohjelmointia kuin mihin tässä työssä on ollut resursseja.

### 5.2.1 Hyvät ja huonot puolet

Xamarinilla saa tehtyä alustalle ominaisen käyttäjäkokemuksen. Xamarinissa käytettävänä ohjelmointikielenä on C#, jonka käytöstä Rimeforgella on ennestään kokemusta. Jotta Xamarinilla saa ohjelmoitua samaan aikaan myös Windows Phone -alustalle, täytyy se integroida Visual Studioon.

Geolokaatio-demosovellus osoitti, että ainakin Androidilla käyttöliittymä on alustalle ominainen. IPhonelle sovellusta ei voitu toteuttaa testilaitteiston puuttumisen vuoksi.

Xamarin on aloittelevan yrityksen käyttöön kallis hankinta, varsinkin jos sen haluaa integroida Visual Studioon. Mikäli yritys ottaa käyttöön indie-hinnoittelumallin, Xamarinia ei voi integroida suoraan Visual Studioon. Tällöin Windows Phonelle tehtävää sovelluskehitystä täytyy tehdä Visual Studiolla rinnakkain Xamarin Studion kanssa. Tämä hidastaa sovelluksen kehitystyötä, on kalliimpaa ja hankaloittaa kehitysversioiden ylläpitoa.

Demosovellus osoitti, että alustariippuvaisesta ohjelmakoodista ei Xamarinin avulla täysin pääse eroon. Sovelluksen käynnistämiseksi, ja että alusta osaisi käyttää Xamarinin kirjastoja, täytyi kutakin alustaa varten käyttää alustan omaa koodia varsinaisen sovelluksen käynnistymiseen asti.

### 5.2.2 Suorat kustannukset

Xamarin Studion hinnoittelu on suoraviivaista. Yrityksille on tarjolla kolme erilaista hinnoittelumallia: indie, business ja enterprise (kuva 5). Kukin hinta on tarkoitettu yhdelle kehittäjälle ja yhdelle alustalle. Alustalla tarkoitetaan tässä yhteydessä alustaa, jolle koodi tehdään. Mikäli yritys haluaa ostaa lisenssin sekä iOSille että Androidille ja integroida ohjelmiston käytön Visual Studioon, täytyy sen ostaa kaksi business-lisenssiä jokaista kehittäjää kohden. Hinta tällöin kehittäjää kohden vuodessa on 1599,40 USD kolmen ensimmäisen vuoden ajan.

Prices below are **per developer, per device platform** with no royalties or hidden fees.

	STARTER <b>FREE</b>	INDIE <b>\$25 / month</b> paid monthly or annually	BUSINESS <b>\$83 / month</b> paid annually (\$999 / year) <small>MOST POPULAR</small>	ENTERPRISE <b>\$158 / month</b> paid annually (\$1899 / year)
Permitted Use	Individual	Individual	Organization	Organization
Subscription Type	N/A	Monthly	Annual	Annual
Deploy to Device	✓	✓	✓	✓
Deploy to App Stores	✓	✓	✓	✓
Xamarin Studio	✓	✓	✓	✓
Unlimited App Size		✓	✓	✓
Xamarin.Forms		✓	✓	✓
Visual Studio Support			✓	✓
Business Features			✓	✓
Email Support			✓	✓
One Business Day SLA				✓
Hotfixes				✓
Technical Kick-off Session				✓
Technical Account Manager				✓
Encrypted Local Data Storage				✓
Code Troubleshooting			At Extra Cost	At Extra Cost
	<a href="#">Download</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>

KUVA 5. Xamarin Studio, hinnat (Simple and transparent pricing 2015)

Yritys saa 20 % alennusta, mikäli se on alle kolme vuotta vanha startup, siinä on alle 21 työntekijää tai siinä on vähintään kolme kehittäjää, jotka haluavat aloittaa Xamarin Studiolla työskentelyn välittömästi. Vuosimaksu yhtä lisenssiä kohden on tällöin 799,20 USD. Tämä hinta on voimassa kolme vuotta Xamarinin käytön aloittamisesta, tai jos yrityksen koko ylittää 20 henkeä. Mikäli yritys maksaa kerralla kolmen vuoden lisenssin, saa se 30 % alennuksen listahinnasta. (Ngo A., Hi Mikko!, liite 1).

### 5.2.3 Käyttöönotto

Jotta Xamarinin voi ottaa käyttöön, täytyy tuote ensin käydä ostamassa Xamarinin verkkokaupasta osoitteesta <https://store.xamarin.com/>. Verkkosivuilta valitaan haluttu

tilausmuoto ja alustat, joita varten Xamarinin haluaa ostaa. Tämän jälkeen sivustoille rekisteröidytään ja syötetään luottokortin maksutiedot. Kun nämä asiat ovat kunnossa, voi tuotteen ladata ja asentaa tietokoneelle. Xamarinin asennuksessa ei ole mitään ihmeellisyyksiä, joten se tapahtuu melko vaivattomasti. Tarkemmat ohjeet löytyvät Xamarinin käyttöönotto-opasteesta liitteestä 4.

### 5.3 Ominaisuuksien vertailu

Toimeksiantajan esittämiä vaatimuksia työkaluille oli kaksi välttämätöntä ja kolme suotavaa. Lisäksi vertailussa piti ottaa huomioon tuotteen hinta. Hinta oli suurin yksittäinen vaikuttava tekijä työkalujen vertailussa.

Välttämättömiä ominaisuuksia työkaluilla olivat alustariippumattomuus ja pääsy APIhin. Suotavia ominaisuuksia työkaluilla olivat tuttu ohjelmointikieli, tuttu ohjelmointiympäristö ja vaivaton testiympäristö. Kaikki muut työkalussa olevat ominaisuudet olivat hyödyksi arvioitaessa työkalun käyttöönottoa. Taulukossa 1 vertaillaan PhoneGapia ja Xamarin Studiota näiden ominaisuuksien osalta.



TAULUKKO 1. Työkaluista löydyttäviä ominaisuuksia

<b>Työkaluista löydyttäviä ominaisuuksia</b>		
<b>Välttämättömät</b>		
	PhoneGap	Xamarin Studio
<b>Alustariippumattomuus</b>	x	x
	PhoneGap on alustariippumaton, sillä voi tehdä sovelluksia 8 eri mobiilille alustalle.	Xamarin on pääosin alustariippumaton, sillä voi tehdä sovelluksia iOSille, Androidille ja Windows Phonelle. Sovelluksien ohjelmoinnissa tarvitaan hieman natiivin koodauksen osaamista.
<b>Pääsy APIhin</b>	x	x
<b>Suotavat</b>		
	PhoneGap	Xamarin Studio
<b>Tuttu ohjelmointikieli</b>	x	x
	Tekniikoina JavaScript, HTML5 ja CSS. Nämä ovat Rimeforgessa tuttuja tekniikoita.	Ohjelmointikielenä C#. C# on Rimeforgessa Unity-käytössä oleva ohjelmointikieli
<b>Tuttu ohjelmointiympäristö</b>	x	½
	Työkaluna voi käyttää mitä tahansa tekstieditoria.	Mikäli ostaa business-lisenssin, Xamarinin voi integroida Visual Studioon.
<b>Vaivaton testiympäristö</b>	x	
<b>Hinta</b>	Ilmainen	799,20 USD / kehittäjä / alusta/ vuosi

## 6 JOHTOPÄÄTÖKSET

Rimeforgen suunnittelemat hyötykäyttöön suunnatut sovellukset on tarkoitettu alustariippumattomiksi. Näiden sovellusten ei tarvitse käytettävyyden tai ulkonäöllisten seikkojen vuoksi tuottaa käyttäjälle kullekin alustalle ominaisia käyttäjäkokemuksia. Sovellusten on tarkoitus näyttää ja tuntua samanlaisilta kaikilla alustoilla. Tällöin käyttäjän on helppo käyttää sovelluksia, vaikka hänen käyttämänsä alustaratkaisu muuttuisi. Lisäksi käyttäjien tukeminen ohjelmiston käytössä helpottuu ohjelman ulkonäön ollessa samanlainen alustasta riippumatta. Rimeforgella kehityksen alla olevat hyötykäyttöön suunnatut mobiilisovellukset eivät vaadi graafisilta ominaisuuksilta paljoa, eikä sovelluksista tule suuria.

PhoneGapin käyttöönottoa puoltaa moni asia. PhoneGap on Apache-lisenssin alainen avoimen lähdekoodin kehys, joka tulee olemaan aina ilmainen. PhoneGapilla on mahdollisuus luoda alustariippumattomia hybridejä sovelluksia pienellä päällekkäisellä työllä. PhoneGapilla on pääsy laitteiden natiiveihin toimintoihin, kuten kameraan, kiihtyvyyssanturiin ja geolokaatioon. PhoneGapia varten voidaan luoda omia laajennuksia omia erityisiä tarpeita varten, ja PhoneGapin aktiivinen kehittäjäyhteisö on luonut paljon valmiita laajennuksia, jotka tarjoavat ylimääräisiä palveluja. PhoneGapille on olemassa Developer App, jolla voi testata omia sovelluksia kohdelaitteissa ilman tarvetta kääntää koodia muutosten jälkeen. PhoneGap ei vaadi Rimeforlessa uuden ohjelmointikielen opettelua. PhoneGapille voi ohjelmoida millä tahansa tekstin editointiin kykenevällä editorilla Notepadista Visual Studioon, jolloin ohjelmointiympäristöön tutustumiseen ei kulu ylimääräistä aikaa. PhoneGapilla tehtyjä sovelluksia voi testata suurella joukolla helposti, ja käyttämällä PhoneGapin sivuilla olevaa hydrate-toimintoa voi sovelluksiin tehtyjä muutoksia kokeilla koko testausjoukolla nopeasti.

PhoneGapin käyttöönottoa vastaan puhuvia argumentteja ei omakohtaisen kokemuksen perusteella, internetistä tai kirjallisuudesta paljoa löytynyt. Sovelluksista voi tulla hitaampia kuin natiivilla koodilla kirjoitetuista sovelluksista. Tämä koskee erityisesti paljon grafiikkaa sisältäviä sovelluksia. Viimeisimpiä alustoja tukevien laajennusten löytäminen voi olla haastavaa, sillä osa näistä laajennuksista on tehty vapaaehtoisvoimin. PhoneGapin ilmaisversiossa sovelluksen suurimmaksi kooksi on rajoitettu 50 MB, ja PhoneGapilla tehdyistä sovelluksista ei tule käyttäjäkokemukseltaan natiivin oloisia.

Xamarinin käyttöönottoa puoltavia asioita on vähäinen määrä. Xamarinilla luodut sovellukset ovat käyttäjäkokemukseltaan natiiveja. Xamarin Studiossa pääasiallisesti käytetty ohjelmointikieli on C#, ja sitä voi käyttää Android- ja iOS-ohjelmointiin. C# on Rimeforgessa pääasiallisesti käytetty ohjelmointikieli. Kun Xamarinin integroi Visual Studioon, voi Visual Studiota käyttämällä ohjelmoida yhtäaikaaisesti Androidille, iOSille ja Windows Phonelle.

Xamarinin käyttöönottoa vastaan puhuvia argumentteja on muutamia. Xamarin Studion käyttö maksaa 25 USD kuukaudessa indie-hinnoittelumallilla käytettävää alustaa ja kehittäjää kohden, eli iOSille ja Androidille ohjelmoitaessa kolmen eri kehittäjän voimin tämän työkalun käyttö maksaa 150 USD kuukaudessa. Mikäli Xamarin Studion haluaa integroida Visual Studioon, maksaa tämän työkalun käyttö kolmelta kehittäjältä iOSille, Androidille ja Windows Phonelle 500 USD kuukaudessa. Xamarinissa joutuu käyttämään alustakohtaista koodia osassa sovellusta, jolloin hyöty alustariippumattomuudesta vähenee. Xamarinin ohjelmointiympäristönä on Xamarin Studio, joten työkaluun tutustumiseen ja tehokkaaseen käyttöönottoon kuluu oma aikansa.

Molemmat tässä opinnäytetyössä tarkastellut työkalut tarjoavat Rimeforgelle samoja asioita, siis alustariippumatonta ohjelmointia. Rimeforgessa hallitaan molemmissa ohjelmistoissa käytetyt ohjelmointikieliet, joten näiden opetteluun ei yrityksessä ohjelmioijilta mene ylimääräistä aikaa. PhoneGapin ohjelmointiympäristö on nopeampaa ottaa tehokkaaseen käyttöön kuin Xamarin Studion. PhoneGapin asennus on kokonaisuutena hieman monimutkaisempi prosessi kuin Xamarin Studion, mikäli alustojen vaatimia SDK:ita ei ole asennettu. Suorilta kustannuksiltaan PhoneGap on edullisempi kuin Xamarin, mikä nuoren yrityksen budjetissa on suuri tekijä.

Mikäli näiden kahden ohjelmointiympäristön välillä täytyy tehdä valinta, suosittelen yritystä sen tämän hetkisen taloudellisen tilanteen huomioon ottaen valitsemaan työkalukseen PhoneGapin.

Tähän opinnäytetyöhön on sisällytetty liitteeksi PhoneGapin-käyttöönotto-opasteen lisäksi myös Xamarin Studion-käyttöönotto-opaste ohjelmiston mahdollista tulevaa käyttöönottoa silmälläpitäen.

## 7 POHDINTA

Työn kirjoittamisen aikana ilmestyi Windows 10, joka tulee laajentamaan Windows-ekosysteemiin kirjoitettavien sovellusten käyttömahdollisuutta. Windows 10:n tarkoituksena on mahdollistaa saman koodin käyttäminen kaikissa Windows 10 -alustoissa. Näitä ovat muun muassa puhelimet, tabletit, tietokoneet ja Xbox-laitteet. Kuinka tämä vaikuttaa PhoneGapin ja Xamarinin käyttömahdollisuuksiin, jää nähtäväksi, mutta näiden asioiden tarkastelu piti rajata työstä pois ajankäytöllisistä syistä.

Koska Xamarin Studio on vielä tässä vaiheessa tilaajalle kallis työkalu käytettäväksi, tuli jo varsin aikaisessa vaiheessa työn tilaajalta toivomus keskittyä enemmän PhoneGapiin. Tämän vuoksi Xamarin Studion tarkastelu on huomattavasti kevyemmin tehty kuin PhoneGapin tarkastelu, ja siihen syvällisemmin tutustuesssa löytyisi vielä paljon enemmän annettavaa kuin mitä tässä työssä annetaan ymmärtää.

Koska Rimeforge käyttää Visual Studiota, nousi opinnäytetyön edetessä esille myös kysymys PhoneGapin käyttöönoton todellisesta tarpeesta. Visual Studiosta ilmestyi työn kirjoittamisen aikana ensin Visual Studio 2015 RC (release candidate) ja pian tämän jälkeen Visual Studio 2015 Community edition. Näissä työkaluissa on mahdollisuus jo asennusvaiheessa ottaa suoraan käyttöön Apache Cordovan tarjoamat laajennukset, jotka ovat samoja laajennuksia kuin mitä PhoneGap käyttää, tosin ilman PhoneGapin tarjoamia lisäosia kuten Developer Appia.

Jatkossa voisi selvittää, kannattaako PhoneGapia käyttää yhtenä lisäympäristönä vai riittääkö tilaajalle Visual Studioon integroitu Apache Cordova, ja millaisia eroja niillä on käytännössä. Lisäksi Xamarin Studioon voisi perehtyä vielä toistaiseksi voimassa olevan opiskelijalisenssin ansiosta paremmin ja selvittää tarkemmin, missä vaiheessa ja millaisilla edellytyksillä tämän työkalun käyttöönotto olisi ajankohtaista tilaajan organisaatiossa.

## LÄHTEET

About Objective-C. Luettu 22.4.2015

<https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>

About Swift. Luettu 22.4.2015

[https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/)

Android Leads Finland's Smartphone Market (but Not by Much). Luettu 21.4.2015

<http://www.emarketer.com/Article.aspx?R=1011346>

Appcelerator. Luettu 2.4.2015

<http://www.appcelerator.com/pricing/>

Application Fundamentals. Luettu 22.4.2015

<http://developer.android.com/guide/components/fundamentals.html>

Choose your plan. Luettu 2.4.2015

<https://build.phonegap.com>

Cross-platform Mobile App Development - Good or Bad, Sumit Garg. Luettu 21.5.2015

<http://tech.co/cross-platform-mobile-app-development-2014-09>

Languages, tools and frameworks. Luettu 21.4.2015

<https://msdn.microsoft.com/en-us/library/windows/apps/dn465799.aspx?f=255&MSPPErrror=-2147217396>

Maximiliano, Firtman, 2013, Programming the Mobile Web, 2nd Edition, O'Reilly Media Inc.

Mobiiliapplikaatioiden kehitys: HTML5, natiivi vai hybridi?. Luettu 22.4.2015

<http://4dsoftware.fi/mobiiliapplikaatioiden-kehitys-html5-natiivi-vai-hybridi/>

Natiivi, hybridi ja HTML5. Luettu 22.4.2015

<http://www.vincit.fi/blog/natiivi-hybridi-ja-html5/>

Ngo, A. Hi Mikko!. Sähköpostiviesti. hello@xamarin.com. Tulostettu 7.4.2015

PhoneGap, Cordova and what's in a name?. Luettu 2.4.2015

<http://phonegap.com/2012/03/19/phonegap-cordova-and-what%E2%80%99s-in-a-name/>

Piotr, J. Kula, 2014, Xamarin Cross-platform Application Development, Packt Publishing.

Sencha. Luettu 2.4.2015

<http://www.sencha.com/>

Shotts, Kerri, 2014, PhoneGap for Enterprise, Packt Publishing.

Simple and transparent pricing. Luettu 7.4.2015

<https://store.xamarin.com/>

Smartphone OS Market Share, Q4 2014. Luettu 21.4.2015

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

Smartphone OS sales market share 2015. Luettu 21.4.2015

<http://www.kantarworldpanel.com/global/smartphone-os-market-share>

Smartphone. Luettu 22.4.2015

<http://www.oxforddictionaries.com/definition/english/smartphone>

Supported Features. Luettu 2.4.2015

[phonegap.com/about/feature](http://phonegap.com/about/feature)

Unity Documentation. Luettu 22.4.2015

<http://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>

Visual Studio. Luettu 22.4.2015

<https://www.visualstudio.com/en-US/products/visual-studio-express-vs>

Wargo, John M., 2014, Apache Cordova API Cookbook, Addison-Wesley Professional.

## LIITTEET

### Liite 1. Sähköposti Xamarinilta

**Anthony Ngo**

APR 07, 2015 | 06:54PM EDT

Hi Mikko,

Thank you for contacting Xamarin! I would like to take this opportunity to introduce myself, my name is Anthony Ngo - I'm a Customer Success Manager in San Francisco, and I will be your personal point of contact for any future licensing questions you might have.

First, I would like to mention that the Indie license is meant for individual and personal use - and if you are planning on using Xamarin for your business, the Business license would better suit you.

With that said, we offer special 20%-off pricing for startups that were founded less than three years ago, and we would be happy to give you that same pricing!

We license on a per platform, per developer basis. As an example, if you have 2 developers that want to develop for both iOS and Android, you will need to purchase 4 licenses.

Therefore, the Business license would be \$799.20 per platform, per developer.

If you are interested in taking advantage of the special pricing, I can create an invoice for you. The invoice will give you 30 days to pay off the licensing cost.

In order to complete your purchase with an invoice, could you please provide me with the following information?

Billing contact name:

Billing contact email:

Company name (if applicable):

VAT (if applicable):

Billing Address:

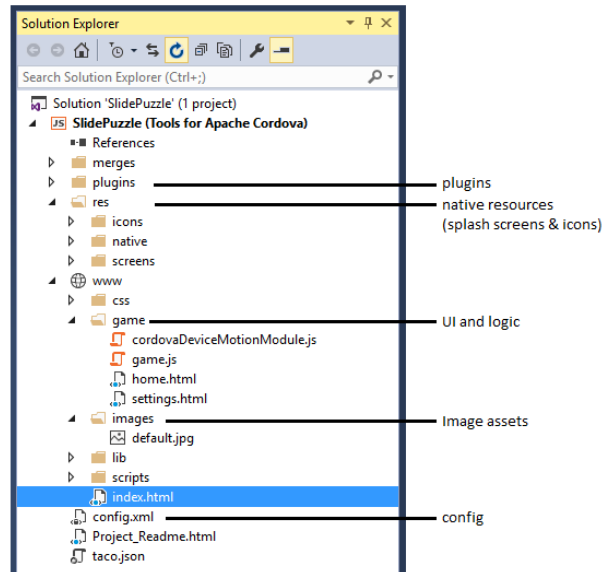
**Liite 2. Visual Studio -projektin muuttaminen PhoneGap projektiksi**

Seuraavilla sivuilla on kuvakaappaukset Microsoftin sivuilta, kuinka Visual Studiolla tehty projekti voidaan muuttaa PhoneGap-projektiksi.



#### Create a new Cordova project using Visual Studio

To create application packages, the cloud-based PhoneGap Build only requires the web-assets of your application, which are limited to your HTML, CSS, images, .js files, and similar files. PhoneGap Build will probably fail to compile your application if native files are uploaded (.h, .m, .java, etc.). Because the project structure created by Visual Studio matches the Cordova project structure, the www folder of your project now hosts all the web assets required by your application.

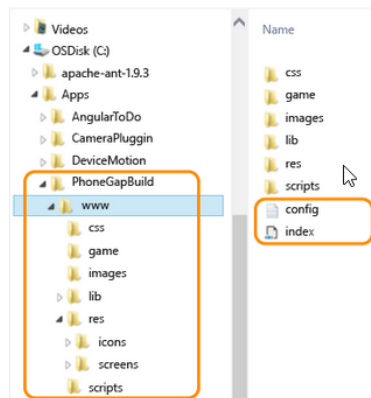


Once you have created a Visual Studio project, you can move your web assets into a PhoneGap Build project. We recommend that you create a new folder (outside your Visual Studio project location) to host your PhoneGap project, and then copy the www folder to this new location.

#### Create the PhoneGap Project Structure

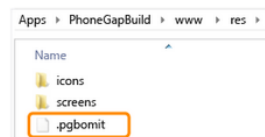
When structuring the PhoneGap project, make sure that config.xml and index.html are in the top level of your app folder structure (here, we use the www folder as the top level folder). Otherwise, you can structure your application as needed. Copy the config.xml from the project's root folder to the www folder of the PhoneGap Build project. Also, copy your native resources from the 'res' folder (res\icons and res\screens respectively) to the www folder of the PhoneGap Build project.

Once you copy over all the files and folders, the PhoneGap Build project should look like this.



Because your application may contain files or folders not required in your final packaged application (such as unused splash screens, bower packages, grunt artifacts, un-compiled less files, etc.), PhoneGap Build supports a special file called .pgbomit.

.pgbomit is a file that you can create and add to a folder to instruct PhoneGap Build to exclude the folder as a source of native application files. (However, you can use this folder to store any files needed during the PhoneGap build process up to the compile step.) As a typical example, you may want to place .pgbomit in the folder containing the icons and splash-screens. So place .pgbomit in the www/res folder of your PhoneGap Build project; as a result, none of files and folders in www/res will be included in the binary app package, except those copied and used for icons and splash-screens for a specific platform. The following illustration shows the .pgbomit file in the www/res folder.



Since PhoneGap Build doesn't support the merges folder of the Cordova CLI project by default, you will need to copy platform-specific contents from the merges folder of your Cordova CLI project into the www folder of the PhoneGap Build project.

## Update Config.xml

PhoneGap Build supports a configuration XML file, config.xml, which is very different from the one generated by Visual Studio project. This configuration file allows you to modify things such as the application's title, icons, splash screens, and other properties.

Start by removing the VS namespace and adding the PhoneGap namespace to the config.xml. The result is shown here.

```
<widget xmlns:cdv="http://cordova.apache.org/ns/1.0"
  id="io.cordova.appb64ec64666e9432a9caf5f01009feb88"
  version="1.0.0.0"
  xmlns:gap="http://phonegap.com/ns/1.0"
  xmlns="http://www.w3.org/ns/widgets">
  <name>SlidePuzzle</name>
```

## Icons

The default icon must be named icon.png and must reside in the root of your application folder. Unless you specify otherwise in config.xml, each platform will try to use the default icon.png during compilation.

```
<icon src="icon.png" />
```

If you want specific icons for the Android platform, the following entries show an example of how to update config.xml based on the contents of your res\icons\android folder in the PhoneGap Build project you previously created.

```
<icon gap:platform="android" gap:qualifier="ldpi"
  src="res/icons/android/icon-36-ldpi.png" />
<icon gap:platform="android" gap:qualifier="mdpi"
  src="res/icons/android/icon-48-mdpi.png" />
<icon gap:platform="android" gap:qualifier="hdpi"
  src="res/icons/android/icon-72-hdpi.png" />
<icon gap:platform="android" gap:qualifier="xhdpi"
  src="res/icons/android/icon-96-xhdpi.png" />
```

For more information on specifying icon elements in config.xml, read this [article](#).

## Splash screens

You can have zero or more splash screen elements present in config.xml. The splash screen element can have src, gap:platform, width and height attributes, just like the <icon> element. Like icon files, save the splash screen files as PNG images. Unless you specify otherwise in config.xml, each platform will use the default splash.png file during compilation. If you do not supply the gap:platform attribute, the default image will be copied to all platforms, increasing the size of each application package.

The default splash screen must be named splash.png and must reside in the root of your application folder.

```
<gap:splash src="splash.png" />
```

If you want specific splash-screens for the Android platform, the following entries show an example of how to update config.xml based on the contents of your res\screens\android folder in the PhoneGap Build project.

```
<gap:splash gap:platform="android" gap:qualifier="port-ldpi"
  src="res/screens/android/screen-ldpi-portrait.png" />
<gap:splash gap:platform="android" gap:qualifier="port-mdpi"
  src="res/screens/android/screen-mdpi-portrait.png" />
<gap:splash gap:platform="android" gap:qualifier="port-hdpi"
  src="res/screens/android/screen-hdpi-portrait.png" />
<gap:splash gap:platform="android" gap:qualifier="port-xhdpi"
  src="res/screens/android/screen-xhdpi-portrait.png" />
<gap:splash gap:platform="android" gap:qualifier="land-ldpi"
  src="res/screens/android/screen-ldpi-landscape.png" />
<gap:splash gap:platform="android" gap:qualifier="land-mdpi"
  src="res/screens/android/screen-mdpi-landscape.png" />
<gap:splash gap:platform="android" gap:qualifier="land-hdpi"
  src="res/screens/android/screen-hdpi-landscape.png" />
<gap:splash gap:platform="android" gap:qualifier="land-xhdpi"
  src="res/screens/android/screen-xhdpi-landscape.png" />
```

For more information on specifying splash screen elements in config.xml, see this [article](#).

## Plugins

To extend access to native platform features exposed by the PhoneGap native-app container, PhoneGap Build supports a white-listed selection of PhoneGap Plugins. For the list of supported plugins, see [Plugins](#). If you include any plugins that aren't in Adobe's white-list, the build will fail. To import a plugin into your PhoneGap Build project, you will need to add the correct `<gap:plugin>` element to `config.xml`. If you omit the version attribute for plugin, the app will always build using the latest version of the plugin.

Here is the most simplistic way of using a versioned plugin.

```
<gap:plugin name="com.phonegap.plugins.example" version="2.2.1" />
```

Because Visual Studio emits the `<vs:plugin>` element into `config.xml` for each plugin you have added, you will need to replace these elements with `<gap:plugin>` elements. In the example project we're using, we use two plugins and two corresponding lines in `config.xml`.

```
<vs:plugin name="org.apache.cordova.device-motion" version="0.2.10" />
<vs:plugin name="org.apache.cordova.camera" version="0.3.2" />
```

In the PhoneGap Build project, these lines must be changed to:

```
<gap:plugin name="org.apache.cordova.device-motion" version="0.2.10" />
<gap:plugin name="org.apache.cordova.camera" version="0.3.2" />
```

For more information on modifying plugins, see this [article](#).

## Upload your project to PhoneGap

Once you have completed all the required modifications, you can upload your application to the PhoneGap build service. First, create an account for the PhoneGap build service. Then, log onto your account at <https://build.phonegap.com/apps> to submit your app. You can submit your application either from a GIT repo or you can upload a local ZIP file. In this example, we are going to upload a local ZIP file which contains the zipped www folder. After you upload the ZIP, you will see that your app is ready to build.

The screenshot shows the 'Your apps' section of the PhoneGap Build service. It displays an app named 'SlidePuzzle' with a version of 1.0.0.0 and App ID 11.73965. The app is owned by user1@fabrikam.com. There are two buttons: a red 'delete' button and a blue 'Ready to build' button.

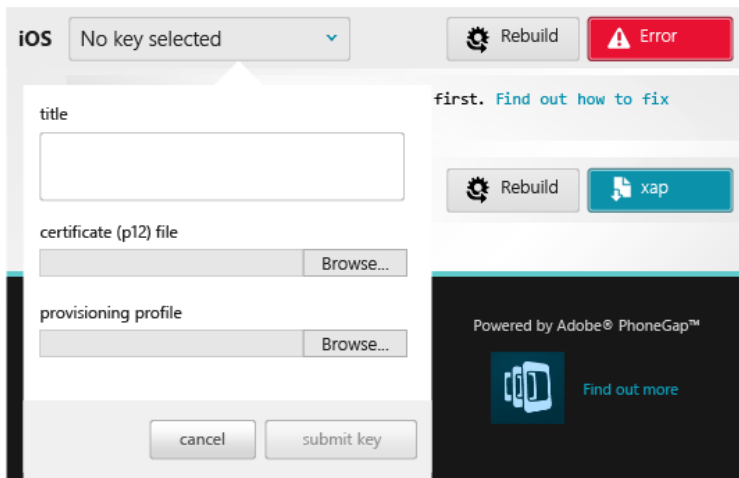
When you click the **Ready to Build** button, the PhoneGap build service will start building for the [platforms that you've defined](#) in `config.xml`. Since we haven't defined any particular platform, the service will build for all the three platforms (iOS, Android, and Windows). After the build completes, you will see the results of the build.

The screenshot shows the build results for the 'SlidePuzzle' app. It features a QR code and an 'Install' button. Below the QR code, there is a dropdown menu showing 'SoakKey' and a 'Rebuild' button. At the bottom, there is a yellow banner asking if the user wants to open or save the file 'SlidePuzzle-release.apk' (1.72 MB) from s3.amazonaws.com. There are 'Open' and 'Sa' buttons.

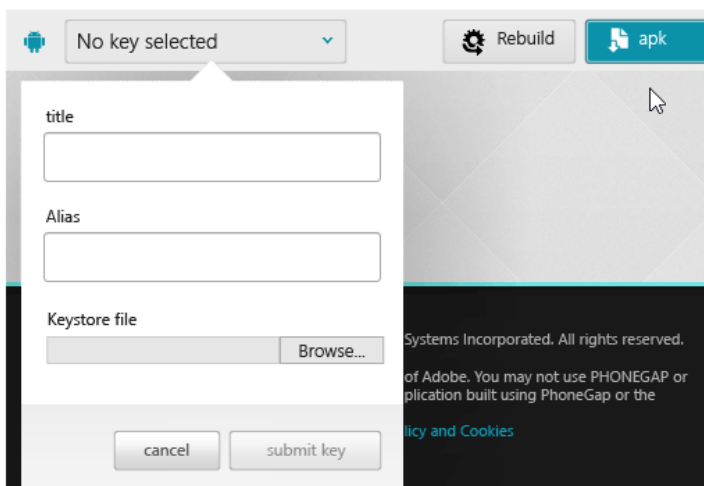
4(4)

#### Code sign and provision your app

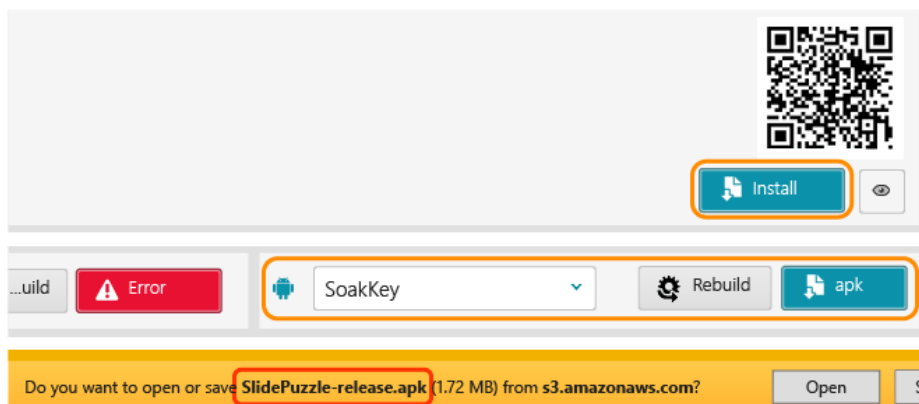
Because we are using a cloud build service, we will have to set up the code signing and provisioning certificates to support building signed release/distribution packages. For iOS, you can provide the code signing certificate and the mobile provisioning profile here:



To build a release APK package for Android that is ready for store submission, PhoneGap build service allows you to provide key-store information and the corresponding passwords here:



For Android, We will use an existing key-store that we have created earlier (or you can create a new one using this [guide](#)) and then rebuild the application. Now, the build service creates a fully signed release package that can either be downloaded for publishing to the store or installed on a tethered device.



We hope this article helps you to convert your Visual Studio Cordova project into a PhoneGap Build project and to quickly build your apps for iOS, Android, or Windows using the PhoneGap build service.

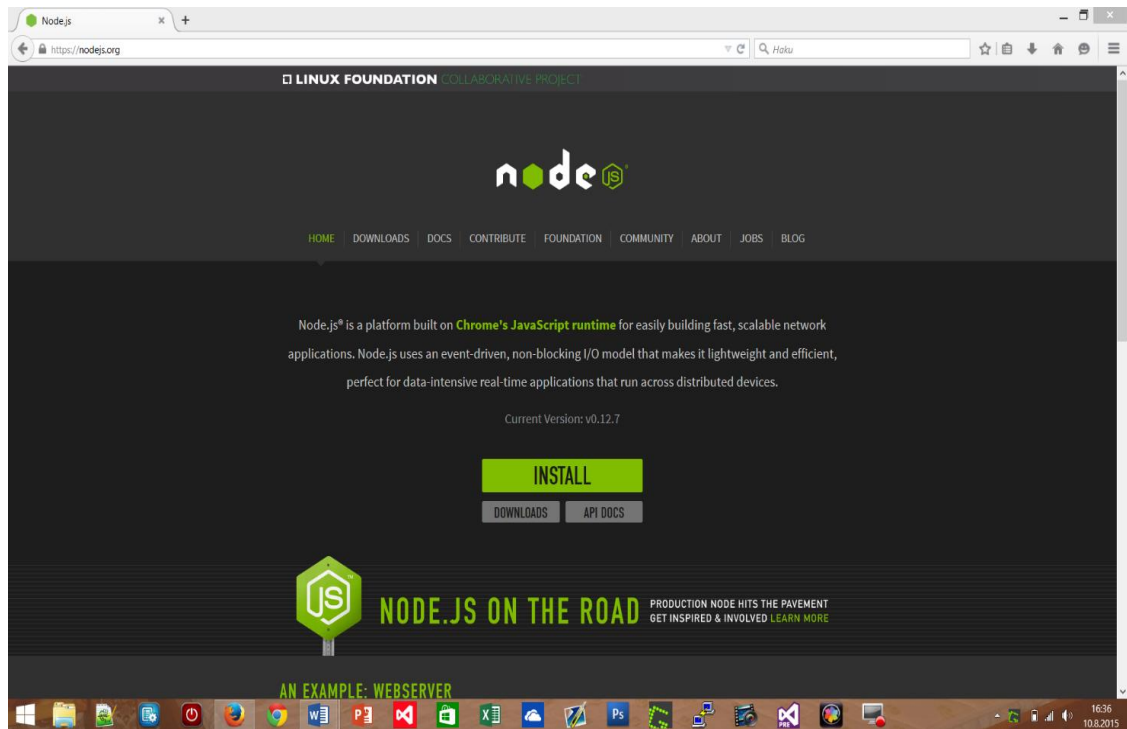
<https://msdn.microsoft.com/en-us/library/dn859242.aspx>(Convert a Cordova Project to a PhoneGap Project).

### **Liite 3. PhoneGap Käyttöönotto-ohjeistus**

1(24)

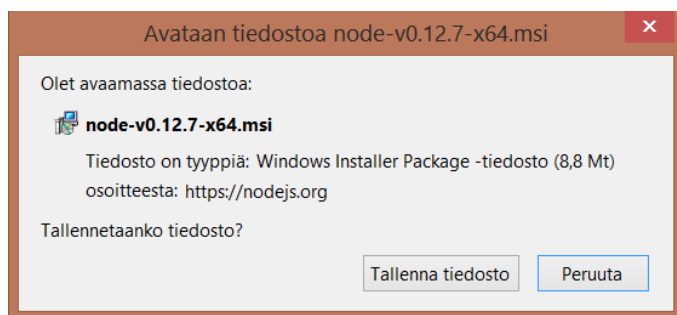
## Node.js asennus

Mikäli koneeseen ei ole vielä asennettu Node.js-ohjelmistoa, se täytyy asentaa ensin. Siirry sivustolle <https://nodejs.org> ja lataa asennuksessa tarvittava tiedosto painamalla install-painiketta keskellä etusivua (kuva 1).



KUVA 1. <https://nodejs.org>, install –kuvake

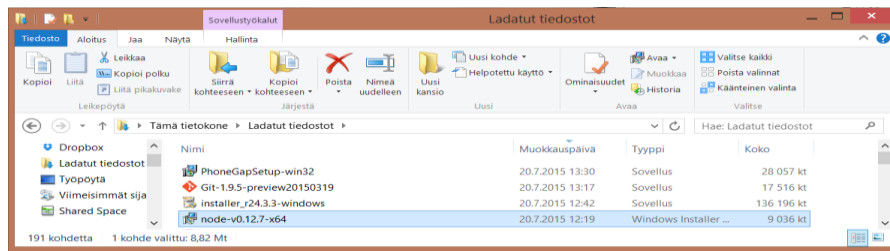
Tallenna install-painikkeen avaama tiedosto painamalla Tallenna tiedosto (kuva 2).



KUVA 2. Node.js, asennusohjelman tallennus

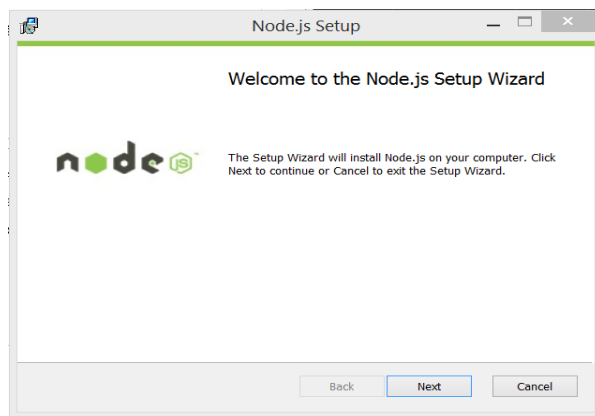
2(24)

Hae ladattujen tiedostojen kansioista äsken tallentamasi tiedosto ja käynnistä se painamalla sitä kahdesti (kuva 3).



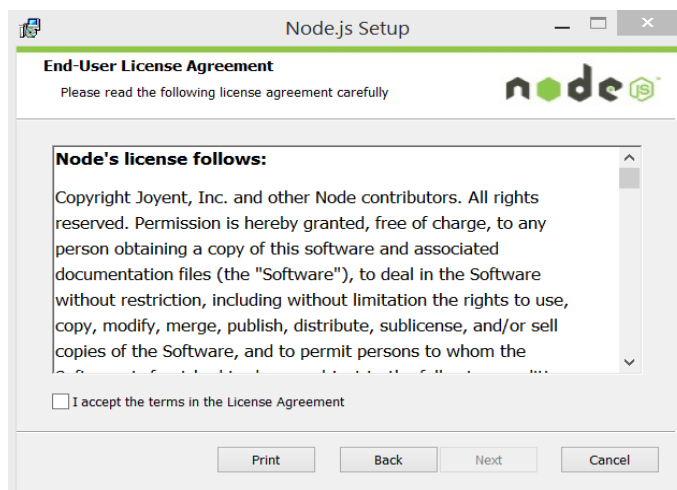
KUVA 3. Node.js, asennusohjelman sijainti

Tällöin Node.js:n asennus käynnistyy. Avautumassa näkyvässä täytyy painaa Next-painiketta, jotta asennus jatkuu (kuva 4).



KUVA 4. Node.js, asennuksen aloitusruutu

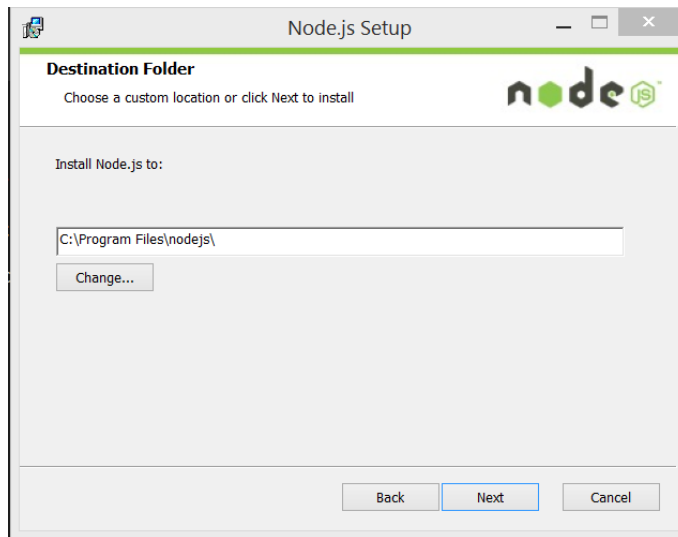
Seuraavassa näytössä hyväksytään Node.js:n käyttöehdot laittamalla ruksi ruutuun kohdassa "I accept the terms in the License Agreement" ja painamalla Next-painiketta (kuva 5).



KUVA 5. Node.js, käyttöehtojen hyväksyminen

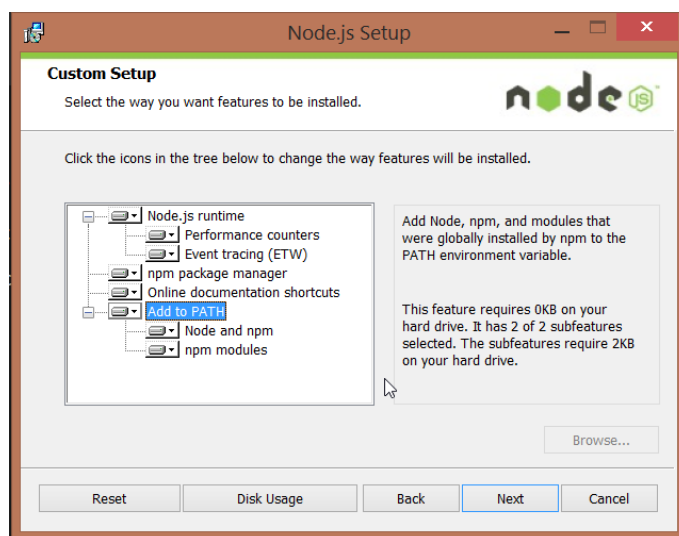
3(24)

Käyttöehtojen hyväksymisen jälkeen asennusohjelmalle kerrotaan, mihin hakemistoon Node.js halutaan asentaa ja painetaan Next-painiketta (kuva 6).



KUVA 6. Node.js, asennushakemiston valitseminen

Seuraavassa valintaruudussa valitaan, kuinka tarvittavat ominaisuudet asennetaan Node.js:n mukana. Oletusarvona on kunkin tarvittavan ominaisuuden asentaminen paikalliselle kiintolevylle, ja näitä oletuksia ei tarvitse muuttaa. Kannattaa tarkastaa, että Add to PATH-kohdassa Node and npm ja npm modules -kohdat on laitettu asentumaan kiintolevylle, tällöin näitä ei tarvitse erikseen manuaalisesti lisätä tietokoneen ympäristömuuttujiin. Tarkistettuasi, että kaikki ominaisuudet asentuvat kiintolevylle, valitse Next (kuva 7).

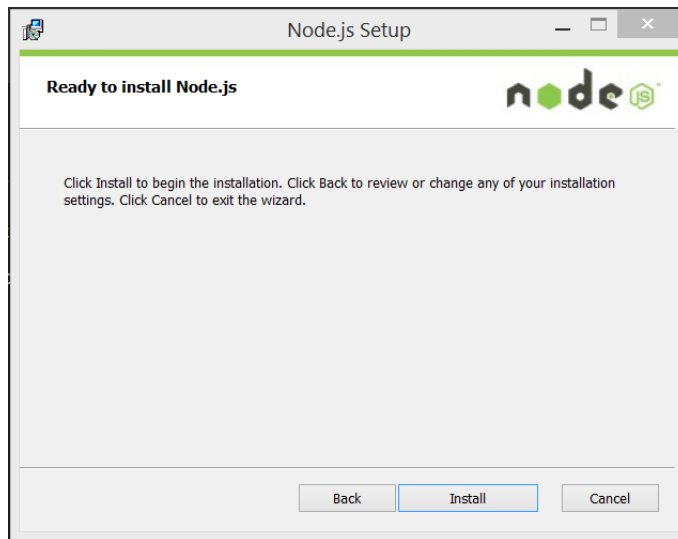


KUVA 7. Ominaisuuksien asennustavan valintavalikko



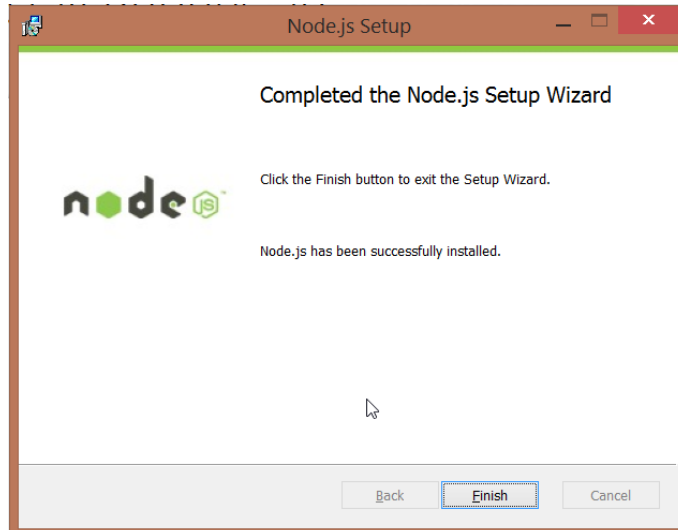
4(24)

Node.js:n asennusohjelma avaa asennuksen vahvistusnäytön. Paina Install ja odottele, että asennusohjelma ilmoittaa asentamisen olevan valmis (kuva 8).



KUVA 8. Node.js, asennuksen vahvistusnäky

Viimeisenä valintaruutuna esille tulee Node.js:n asennuksen valmistumisnäky. Paina Finish. Tietokonetta ei tarvitse käynnistää uudelleen (kuva 9).



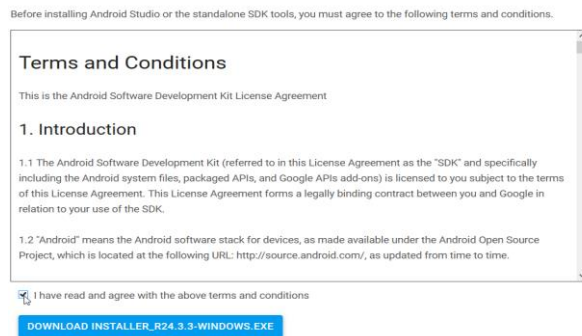
KUVA 9. Node.js, asennuksen valmistumisen ilmoitusruutu

Tässä kappaleessa opastettiin PhoneGapin käyttöönotossa tarvittavan Node.js ohjelman asennus PC-tietokoneelle. Seuraavassa kappaleessa käydään läpi Android SDK:n asennus.

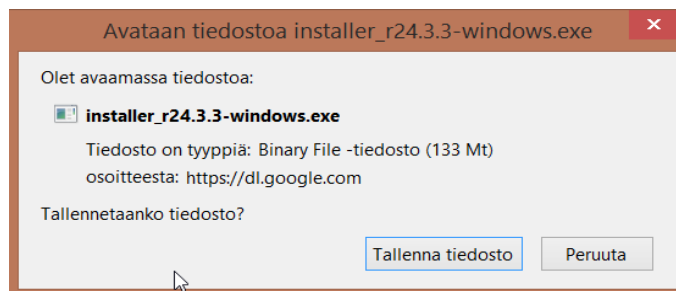
## Android SDK:n asennus

Mikäli koneeseen ei ole vielä asennettu Android SDK:ta, täytyy se asentaa koneelle. Tässä kappaleessa käydään läpi Android SDK:n asennus. Siirry sivustolle <http://developer.android.com/sdk/index.html#Other> ja lataa sieltä uusin SDK tools only -paketti. Tätä kirjoittaessa paketin nimi on installer\_r24.3.3-windows.exe. Joudut suostumaan käyttöehtoihin, ennen kuin voit aloittaa paketin lataamisen (kuvat 10 ja 11).

### Download

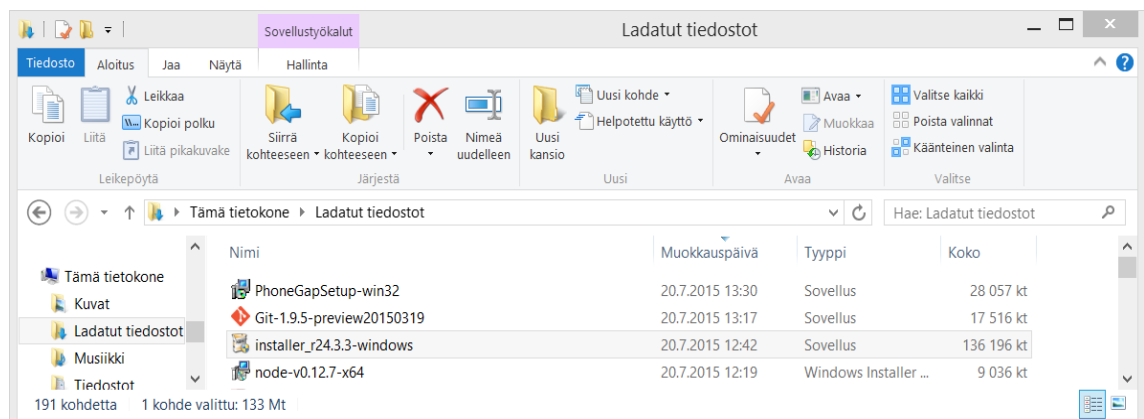


KUVA 10. Android SDK Tools, lataaminen ja suostuminen käyttöehtoihin



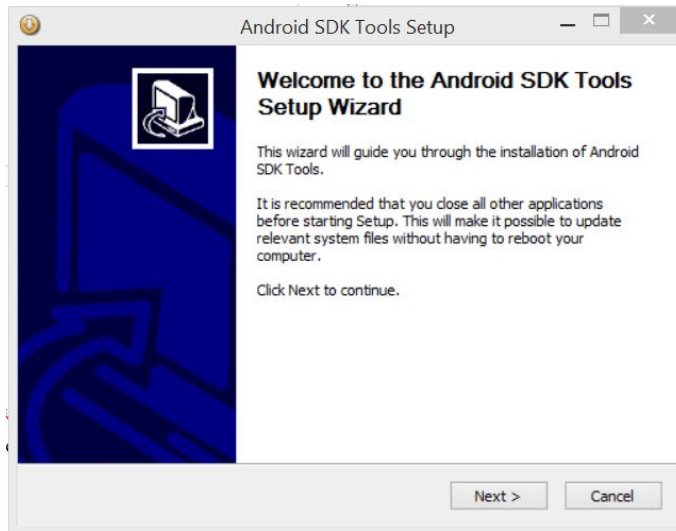
KUVA 11. SDKN:n asennuspaketin tallennusruutu

Siirry ladattujen tiedostojen kansioon, paikanna sieltä kyseinen ohjelma ja käynnistä se (kuva 12).



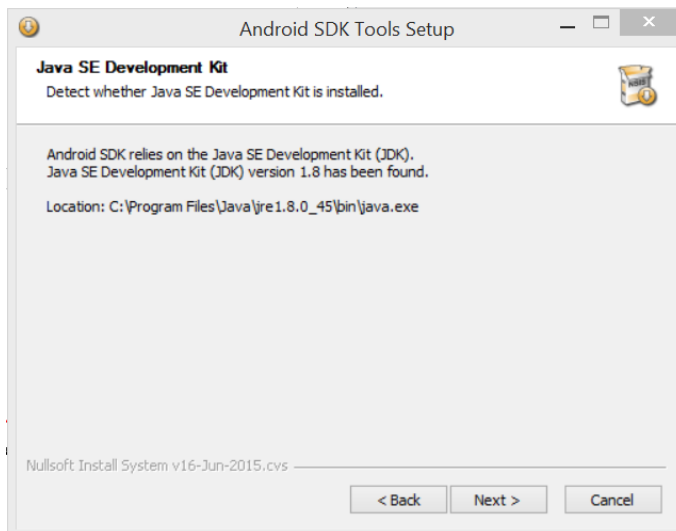
KUVA 12. Android SDK, asennuspaketti

Ohjelman käynnistäminen avaa kuvan 13 mukaisen näkymän. Tästä näkymästä valitaan Next.



KUVA 13. Android SDK, asennuksen aloitusvalikko

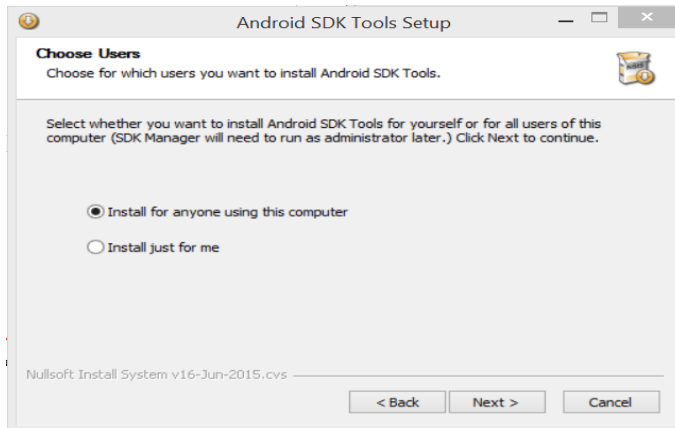
Seuraavassa valintaruudussa asennusohjelma tarkastaa, onko koneelle asennettu Java SE Development Kit. Tässä käyttöönotto-opasteessa oletetaan, että kyseinen ohjelmisto on jo asennettu koneelle. Tarkastuksen päättymisen jälkeen painetaan Next-painiketta (kuva 14).



KUVA 14. Java SE Development Kit, asennuksen tarkastus

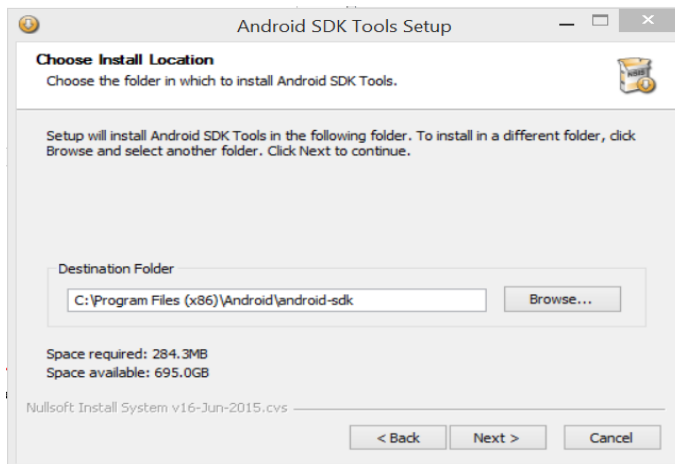
7(24)

Seuraavassa ruudussa valitaan, halutaanko SDK asentaa vain omaan käyttöön vai kaikille konetta käyttäville henkilöille. Valitse Install for anyone using this computer ja paina Next (kuva 15).



KUVA 15. SDK Tools, asennuksen valinta, kenen käyttöön ohjelmisto asennetaan

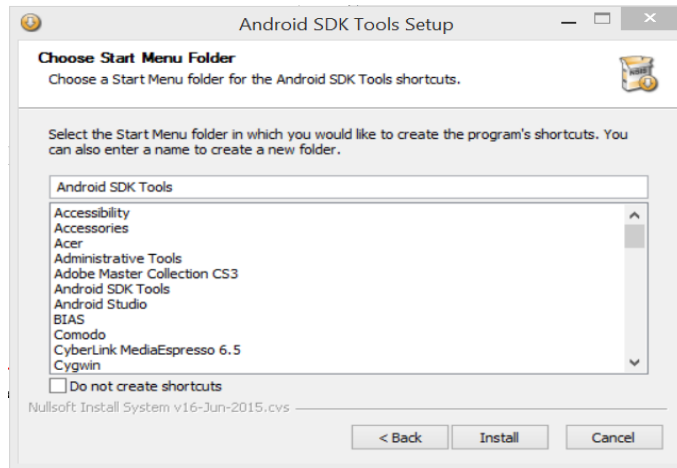
Seuraavaksi valitaan mihin, hakemistoon ohjelmisto halutaan asentaa. Valitse Next (kuva 16).



KUVA 16. Asennusvalikon valinta

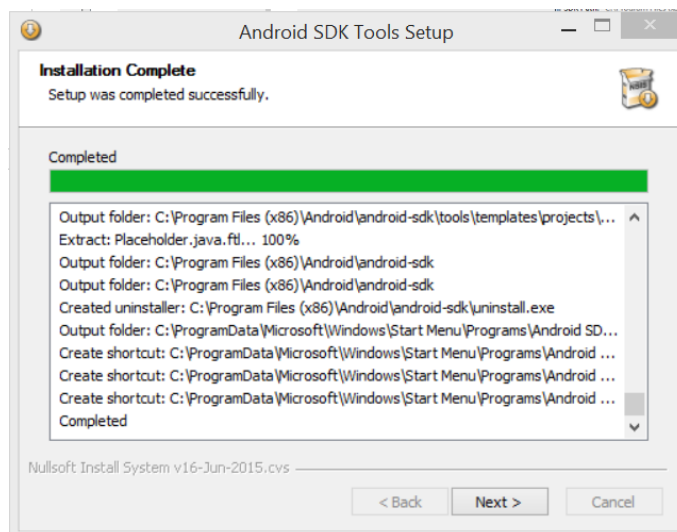
8(24)

Asennusvalikon valinnan jälkeen asennusohjelma ehdottaa pikalinkin luomista Aloitusvalikkoon. Oletusarvoisesti pikalinkki luodaan nimellä Android SDK Tools. Oletusarvo on tarpeeseen soveltuva, joten valitse Install (kuva 17).



KUVA 17. Pikalinkin luominen

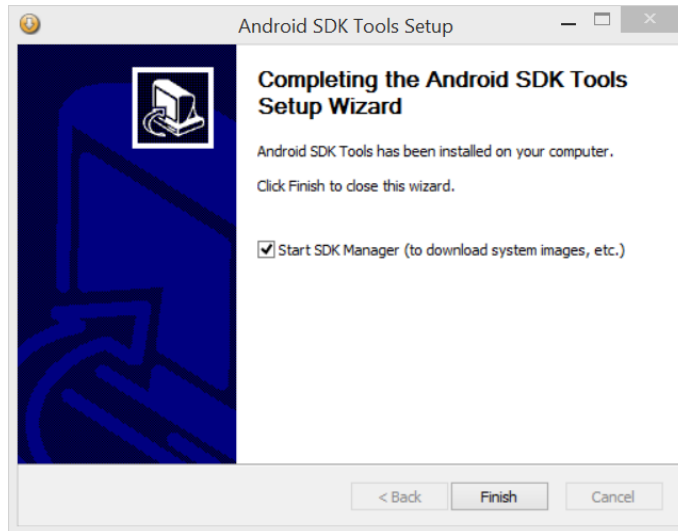
Asennusohjelman asennettua Android SDK Tools:n onnistuneesti tulee asennuksen etenemisen seurantaan näyttöön ilmoitus Installation Complete. Paina Next (KUVA 18).



Kuva 18. Asennuksen tilan seuranta näyttö

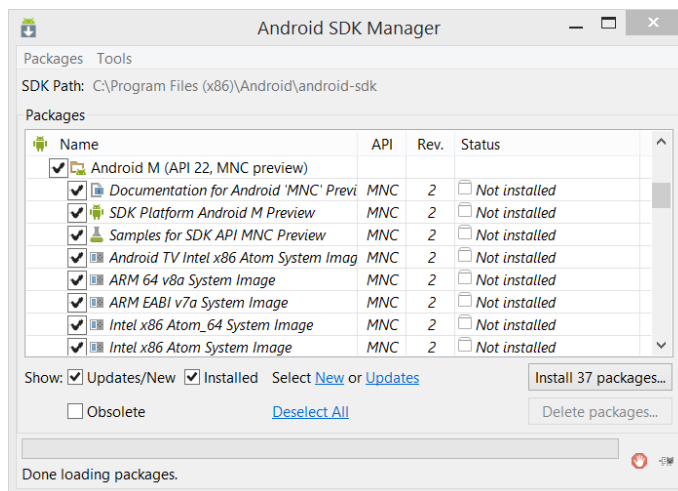
9(24)

Asennuksen valmistuttua asennusohjelma vielä ilmoittaa, että Android SDK Tools on asennettu koneellesi. Seuraavaksi asennusohjelma ehdottaa SDK Managerin käynnistämistä, jotta koneelle voidaan asentaa Android-ohjelmoinnissa tarvittavat alustat. Anna kyseisen valintaruudun olla valittuna ja paina Finish (kuva 19).



KUVA 19. Android SDK Tools, asennuksen lopetus ja SDK Managerin käynnistys

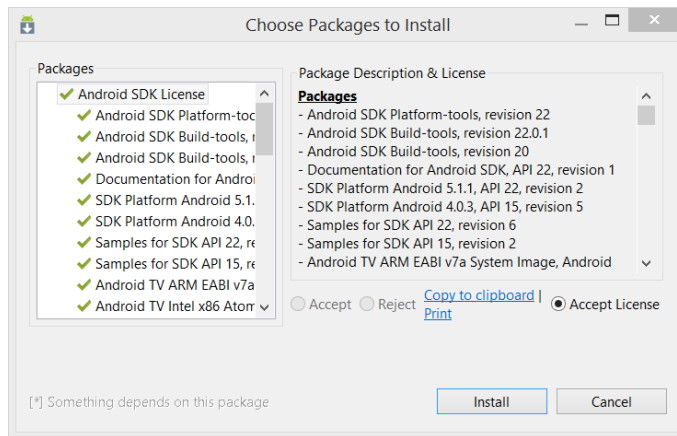
Android SDK Manager käynnistyy ja ehdottaa joidenkin Android-alustojen lataamista. Valitse SDK Managerista haluamasi alustat, eli API:t ja paina Install nn packages (jossa nn on valitsemiesi APIen ym. lukumäärä)(kuva 20).



KUVA 20. Android APIt, valintaruutu

10(24)

Kun olet valinnut haluamasi API:t ja painanut Install -nappia, näytölle tulee vielä varmistusruutu. Valitse Accept License kohta ja paina Install (kuva 21).



KUVA 21. APIen asennuksen varmistus

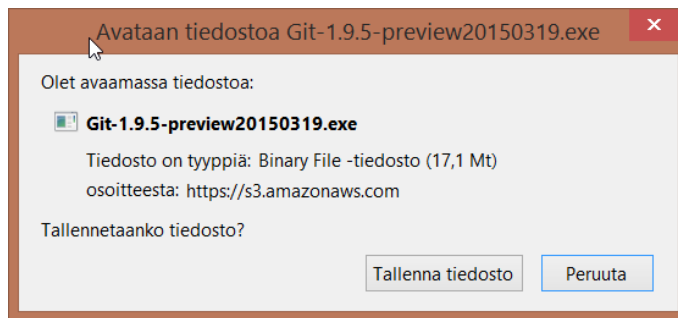
SDK Manager etsii, lataa ja asentaa valitut API:t, minkä jälkeen ohjelma ilmoittaa onnistuneiden asennusten määrän. Asennusprosessiin menee oma aikansa, varsinkin jos olet valinnut useamman API:n asennettavaksi. Onnistuneen asennuksen jälkeen voit klikata näytöltä Close ja sulkea SDK Managerin.

Tässä kappaleessa opastettiin PhoneGapin käyttöönotossa tarvittavan Android SDK Managerin ja APIen asennus. Seuraavassa kappaleessa käsitellään Git-ohjelmiston asennus.

## Git Asennus

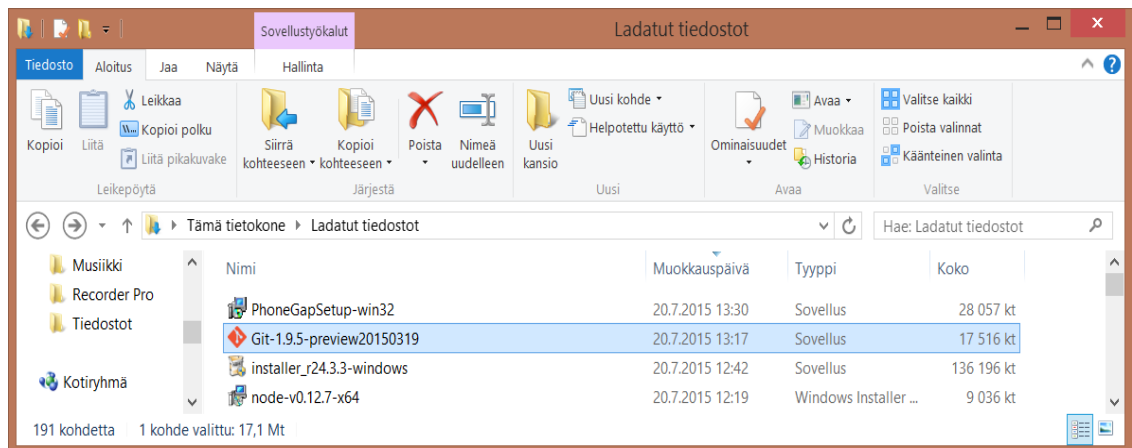
PhoneGapin käytössä tarvitaan Git-versionhallintajärjestelmää. Tässä kappaleessa käydään läpi Gitin asentaminen.

Siirry sivustolle <http://git-scm.com/download/win>, ja mikäli Gitin asennustiedoston lataaminen ei käynnisty automaattisesti, paina linkkiä [click here to download manually](#). Näytölle avautuu ruutu, jossa kehoitetaan tallentamaan Gitin asennustiedosto. Tallenna tiedosto (kuva 22).



KUVA 22. Git-asennustiedoston tallennus

Mene hakemistoon, jossa tallentamasi tiedosto sijaitsee, ja käynnistä ohjelma tuplaklikkaamalla sitä (kuva 23).

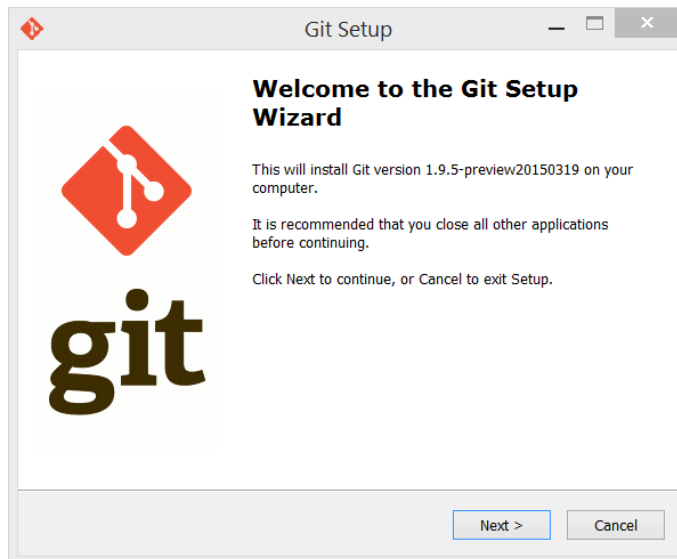


KUVA 23. Git-asennustiedoston sijainti



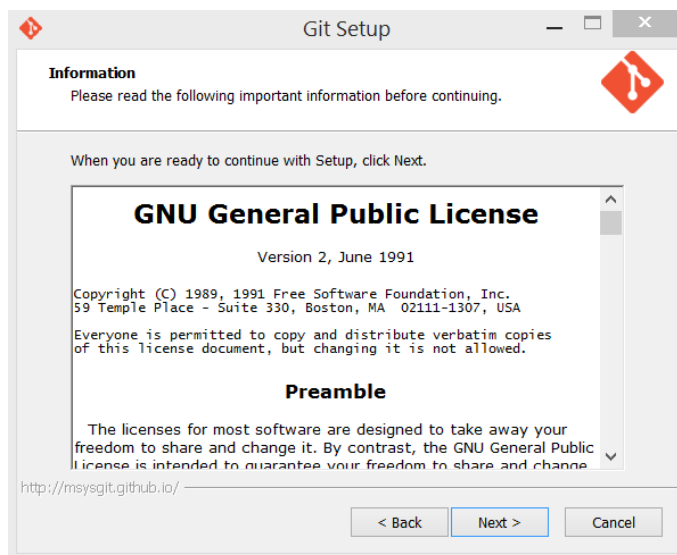
12(24)

Ohjelma avaa asennuksen aloitusnäytön, jossa kehoitetaan sulkemaan kaikki taustalla toimivat ohjelmat. Paina ruudulla näkyvää Next-painiketta (kuva 24).



KUVA 24. Git-asennuksen aloitusnäyttö

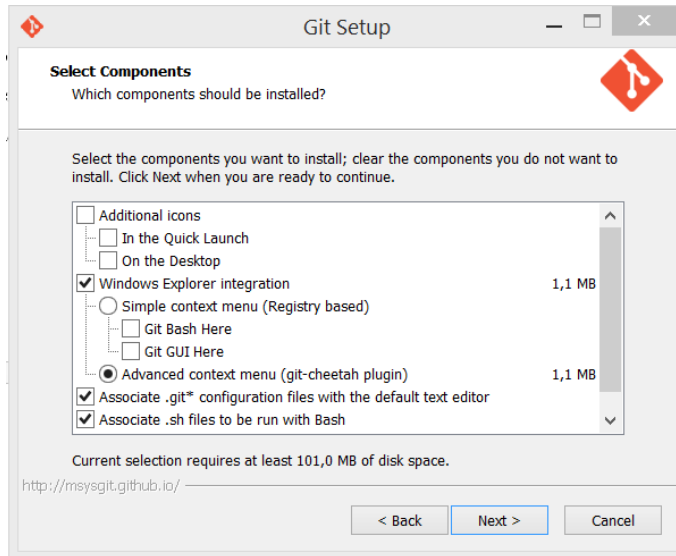
Seuraavalla ruudulla pyydetään lukemaan Gitin käyttöehdot, GNU General Public License versio 2. Luettuasi, tai mikäli tiedät mitä tämä lisenssi sisältää, paina Next (kuva 25).



KUVA 25. Git Setup, GNU General Public License

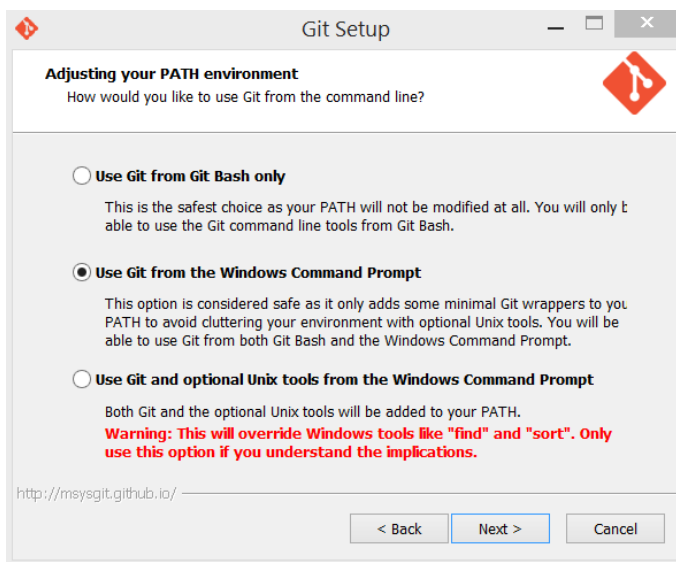
13(24)

Luettuasi ohjelmiston käyttöehdot avautuu näkymä, jossa valitaan mitä komponentteja haluat asentaa. Oletusarvoisesti asennusohjelma tarjoaa kuvan 26 mukaisia valintoja. Pidä valinnat oletusarvoisina ja paina Next.



KUVA 26. Asennettävien komponenttien valinta

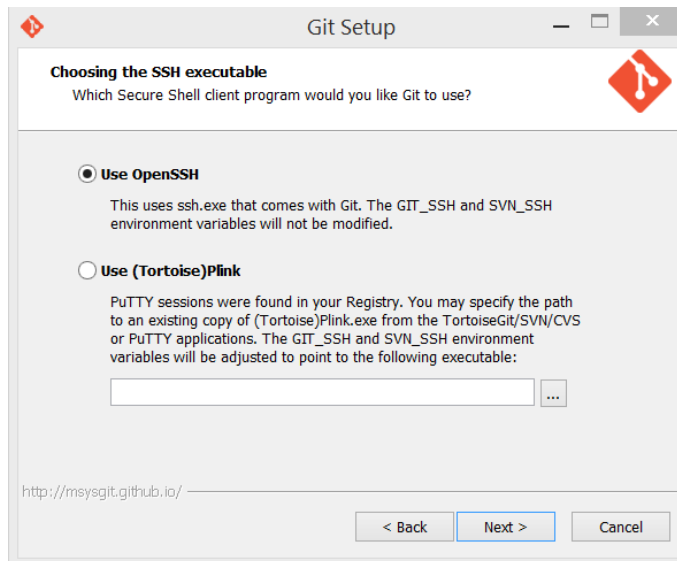
Seuraava valintaruutu kysyy, kuinka haluat käyttää Gitiä komentoriviltä. Valitse keskimäinen, Use Git from the Windows Command Prompt. Tällöin asennusohjelma muokkaa PATH-ympäristömuuttujaa siten, että voit käyttää Gitiä mistä tahansa komentorivikehotteen hakemistosta. Paina Next (kuva 27).



KUVA 27. Ympäristömuuttujien muokkaaminen Gitin asennuksessa

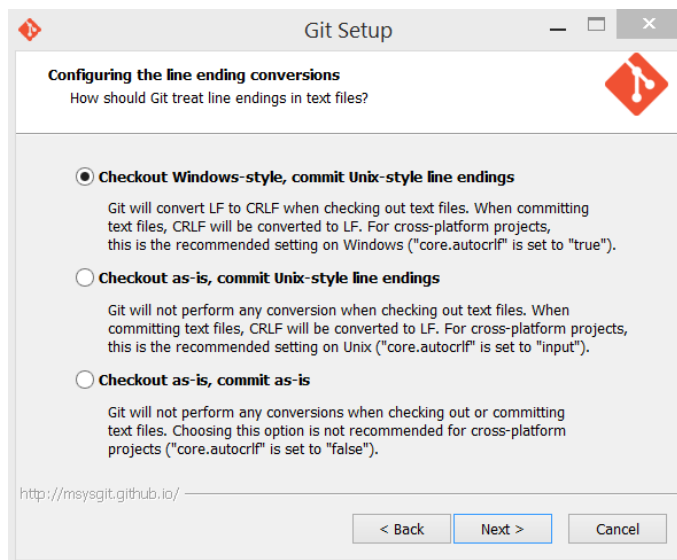
14(24)

Asennuksen seuraavassa vaiheessa sinulta kysytään, mitä Secure Shell client -ohjelmaa haluat Gitin käyttävän. Valitse Use OpenSSH radiobutton ja paina Next (kuva 28).



KUVA 28. SSH:n valinta

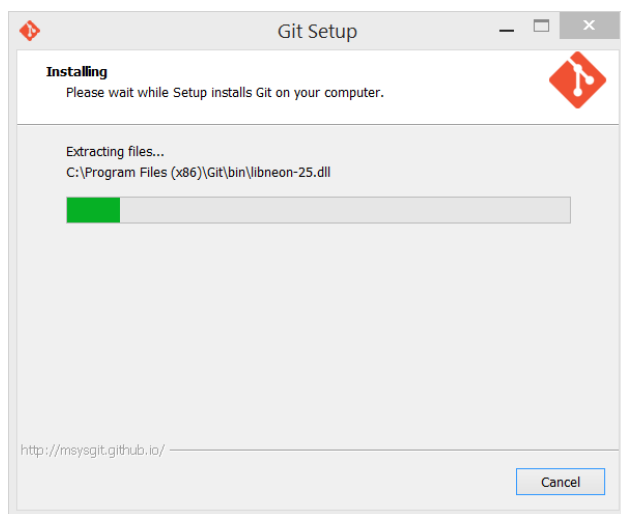
Seuraavassa vaiheessa valitaan, kuinka haluat Gitin kohtelevan rivien päätteitä tekstitiedostoissa. Anna radiobuttonin valinnan olla Checkout Windows-style, commit Unix-style line endings -valinnan päällä. Tämä on asennusohjelman suosittelema valinta Windows-ympäristössä. Valitse Next (kuva 29).



KUVA 29. Rivien päättymisen kohtelu tekstitiedostoissa Gitissä

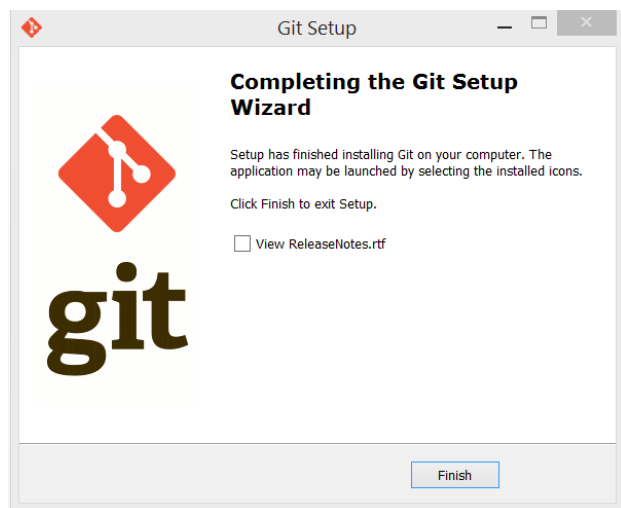
15(24)

Painettuasi Next asennus alkaa välittömästi ilman, että sinulta kysytään vahvistusta tekemiisi valintoihin. Ole siis varma, että olet tehnyt oikeat valinnat (kuva 30).



KUVA 30. Gitin asennus.

Onnistuneen asennuksen jälkeen Git Setup tuo vielä esille näytön, jossa voit valita haluatko lukea Release Notes -tiedostoa. Halutessasi jätä valintaruutu valituksi ja paina Finish (kuva 31).



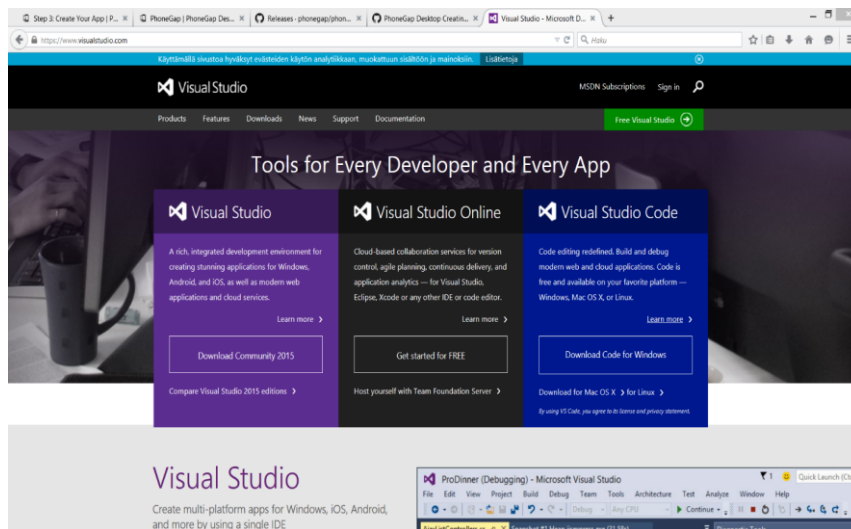
KUVA 31. Git asennuksen viimeistely

Tässä kappaleessa opastettiin PhoneGapin käytössä tarvittavan Git-versionhallintajärjestelmän asentaminen Windows-ympäristöön. Seuraavassa kappaleessa käydään läpi Visual Studio Community 2015 -ohjelmiston ja sitä kautta Windows Phone SDK:n asennus.

## Visual Studio asennus

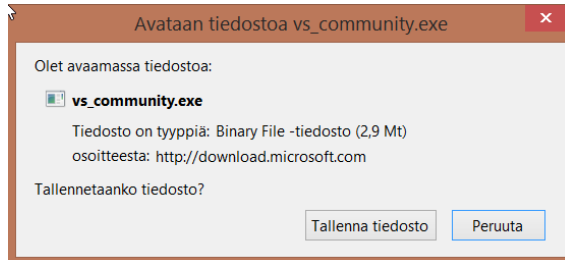
Jotta ohjelmisto toimisi Windows Phonella, tarvitsee koneelle asentaa myös Windows Phone SDKt. Näiden asentaminen käy helpoiten asentamalla Visual Studio Community 2015. Tämän ohjelmiston saavat asentaa yksityishenkilöt omien sovellusten myyntiin tai muihin tarkoituksiin. Yrityksessä, jossa on vähemmän kuin 250 PC:tä tai alle miljoonan USD:n vuosittainen liikevaihto, on viidellä yksittäisellä käyttäjällä lupa käyttää Visual Studio Community 2015 editionia yhtä aikaa sovellusten kehittämiseen ja testaamiseen. Visual Studiota ei kuitenkaan ole pakko käyttää sovellusten kehittämiseen jos käyttää PhoneGapia, jolloin ei Visual Studion lisenssistä tarvitse olla huolissaan.

Asentaaksesi Visual Studio Community 2015:n mene osoitteeseen <https://www.visualstudio.com/> ja paina sivulla olevaa linkkiä Download Community 2015 (kuva 32).



KUVA 32. Visual Studio Community 2015 lataussivu

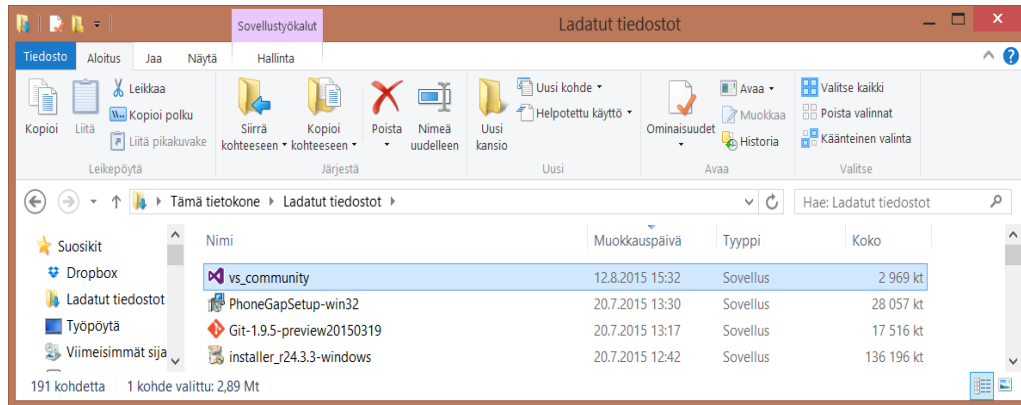
Tallenna linkistä avautuva tiedosto koneellesi painamalla Tallenna tiedosto (kuva 33).



KUVA 33. Tallenna vs\_community.exe

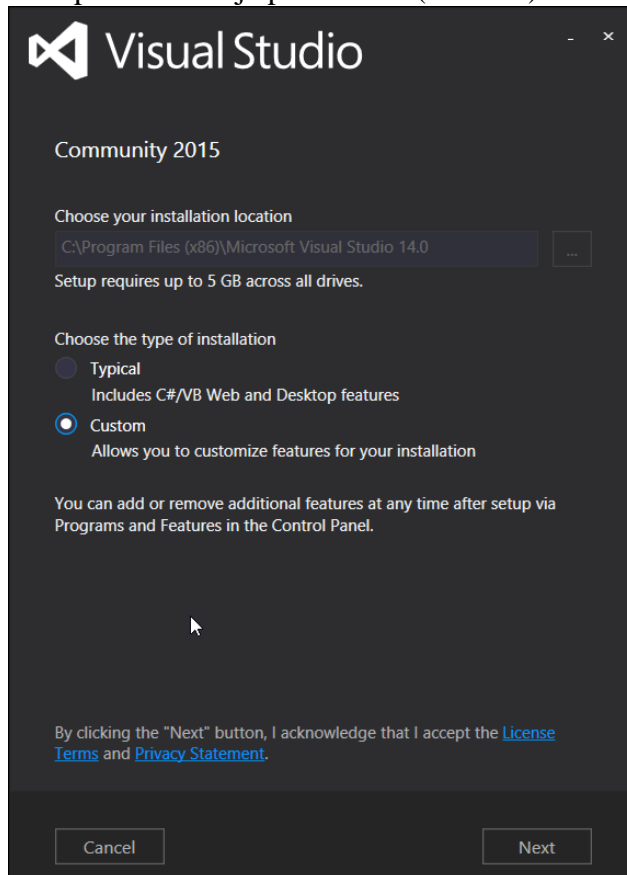
17(24)

Avaa tiedosto hakemistosta, johon oletusarvoisesti tallennat ladatut tiedostot (kuva 34).



KUVA 34. vs\_community tallennussijainti

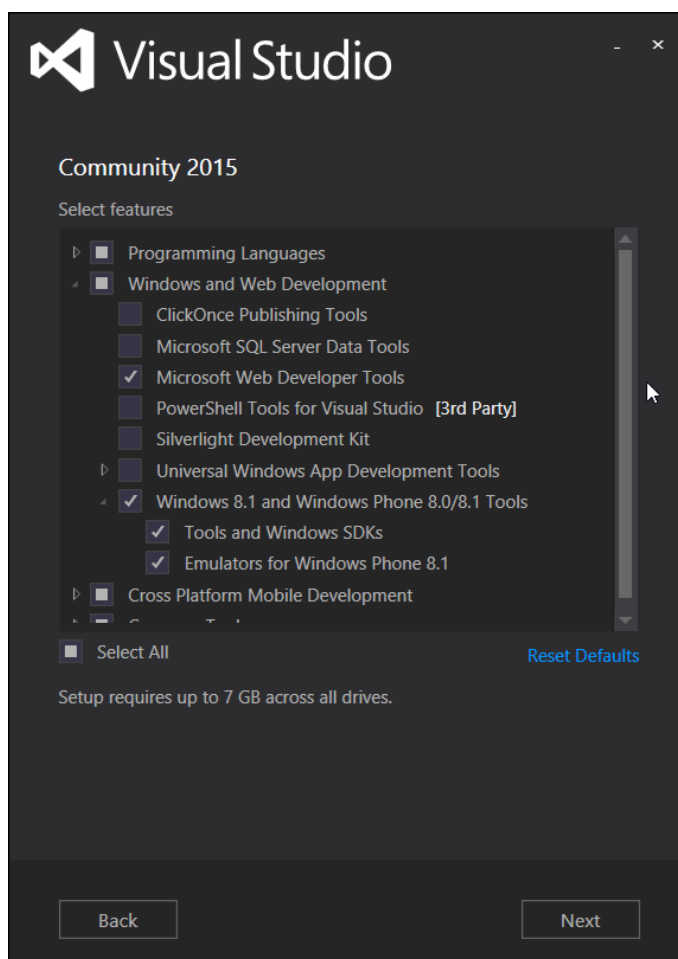
Visual Studio Community 2015:n asennuksen käynnistyttyä valitse Custom asennus radiopainikkeista ja paina Next (kuva 35).



KUVA 35. VS Community 2015 Custom asennus

18(24)

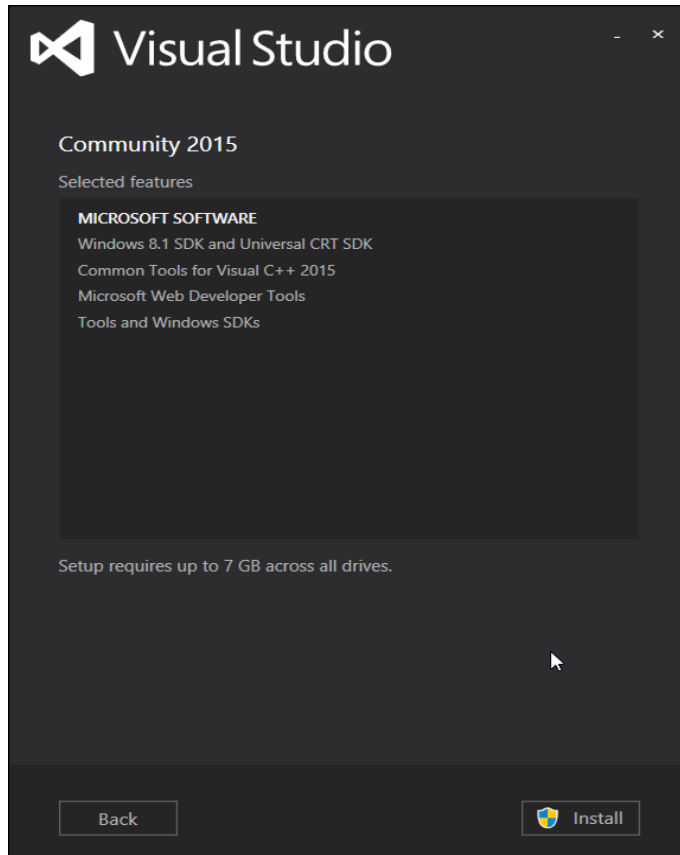
Asentaessasi VS Community 2015 -ohjelmaa, tarkasta Custom-asennustavalla, että Windows and Web Development -valikon alta on valittuna Windows 8.1 and Windows Phone 8.0/8.1 Tools -kohdasta vähintään Tools and Windows SDKs (kuva 36). Tätä kirjoittaessa on juuri julkaistu Windows 10, joten asennukseen kannattaa ottaa mukaan myös Universal Windows App Development Tools. Halutessasi voit asentaa koko VS Community 2015 -paketin, jolloin asennuksen koko on 22 gigatavua, mutta mukana tulee liitännäinen Apache Cordovaan, Android-kehittämisessä tarvittavat työkalut, Visual C++, Visual F# ja Python-kielet ja paljon muuta. Valittuasi haluamasi toiminnot asennusta varten paina Next.



KUVA 36. VS Community 2015 ja WP SDK asennus

19(24)

Seuraavaksi näkyviin tulee tarkistusruutu, jossa voit tarkastaa että kaikki haluamasi toiminnot tulevat mukaan asennukseen. Mikäli olet tyytyväinen tekemiisi valintoihin, paina Install (kuva 37).

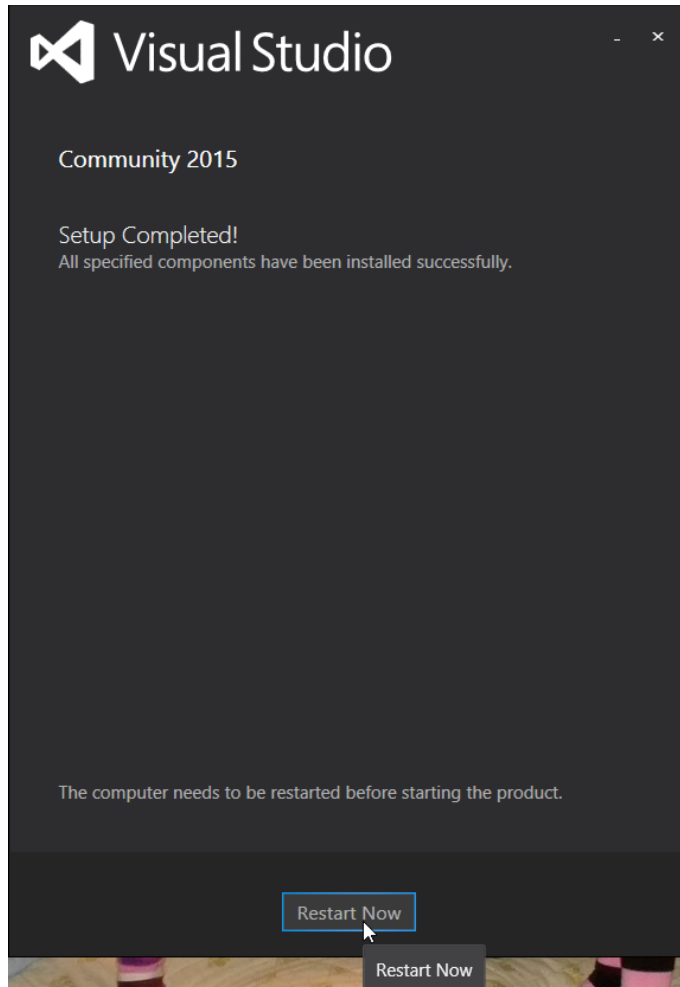


KUVA 37. Visual Studio Community 2015 asennettavat ominaisuudet



20(24)

Onnistuneen asennuksen jälkeen asennusohjelma ilmoittaa asennuksen päättyneen. Ohjelmisto voidaan ottaa käyttöön tietokoneen uudelleenkäynnistyksen jälkeen. Paina Restart Now (kuva 38).



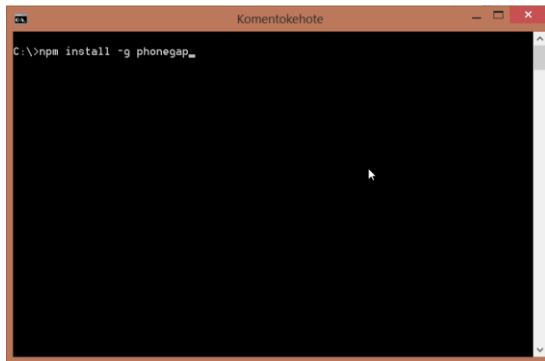
KUVA 38. Visual Studio Community 2015 valmis asennus

Asentamalla Visual Studion saa samalla käyttöönsä Windows Phone SDK:n, jota tarvitaan että pystyy kehittämään sovelluksia Windows Phonelle. Seuraavassa kappaleessa käydään läpi itse PhoneGap-ohjelmiston asennus.

## PhoneGapin asennus

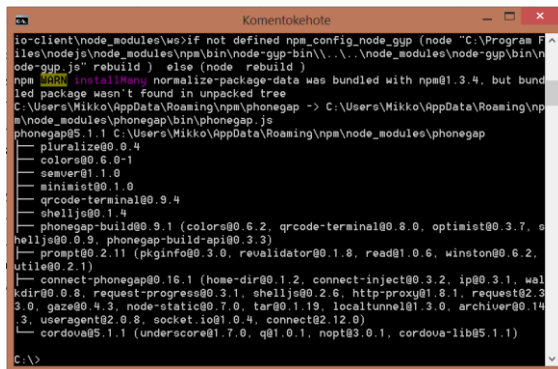
Tässä kappaleessa käydään läpi PhoneGapin asennus. PhoneGap vaatii toimiakseen tässä käyttöönotto-opasteessa jo opastettujen ohjelmistojen asentamisen. Mikäli olet jo asentanut nämä ohjelmistot onnistuneesti, voit asentaa PhoneGapin.

Avaa Windowsin komentokehote CMD ja kirjoita konsoliin komento `npm install -g phonegap` ja paina enter (kuva 39).



KUVA 39. PhoneGap asennuksen aloittava komento

Tämän jälkeen PhoneGap hakee, lataa ja asentaa PhoneGap ohjelmiston koneellesi (kuva 40).



KUVA 40. PhoneGap asennus valmis

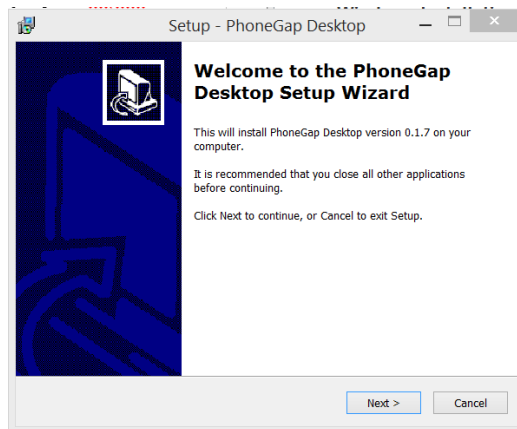
PhoneGapin komentoriviversio on nyt asennettu koneellesi, ja PhoneGap on käyttövalmiina.

Tässä kappaleessa käytiin läpi PhoneGapin komentoriviversion asentaminen Windowsympäristöön. Mielestäni PhoneGapin työpöytäversio helpottaa ohjelmiston käyttöä ohjelmoitavan ohjelman testivaiheessa. Tämän vuoksi käyn seuraavassa kappaleessa läpi PhoneGapin työpöytäversion asentamisen, vaikka PhoneGap on jo näillä ohjeilla valmis käytettäväksi.

## PhoneGap Desktop -ohjelmiston asennus

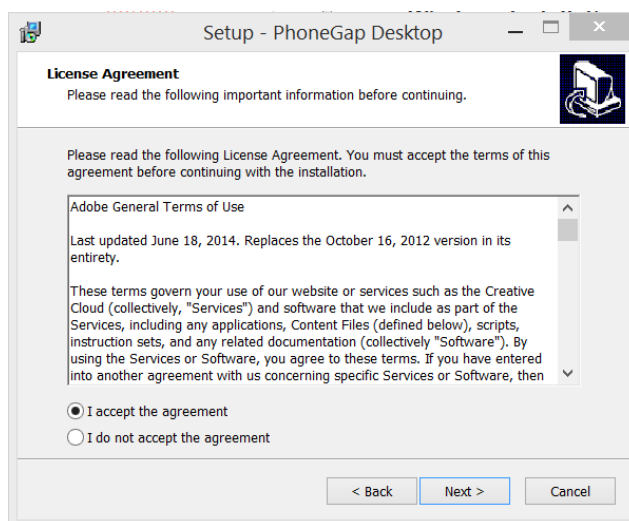
Tässä kappaleessa käydään läpi PhoneGap Desktop -ohjelmiston asennus. Desktop versio yhdessä puhelimeen asennettavan PhoneGap Developer appsin kanssa helpottaa PhoneGapilla ohjelmoitavan sovelluksen testausta.

Siirry osoitteeseen <http://docs.phonegap.com/getting-started/1-install-phonegap/desktop/> ja hakeudu PhoneGapin Windowsin asennusohjeiden kohdalle. Lataa ja tallenna koneellesi uusin versio PhoneGap Desktopista (kirjoittamisen hetkellä versio 0.1.8). Paikanna juuri tallentamasi tiedosto ladatuista tiedostoista ja käynnistä se. Näytölle avautuu PhoneGap Desktop Setup Wizard (kuva 41). Paina Next.



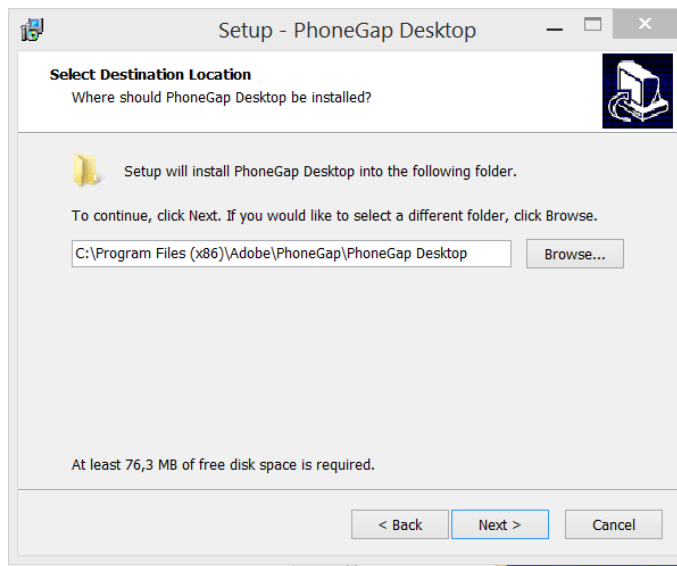
KUVA 41. PhoneGap Desktop Setup Wizard

Seuraavassa ruudussa kerrotaan PhoneGap Desktopin käyttöehdot. Mikäli haluat asentaa ohjelmiston koneellesi, hyväksy käyttöehdot ja paina Next (kuva 42).



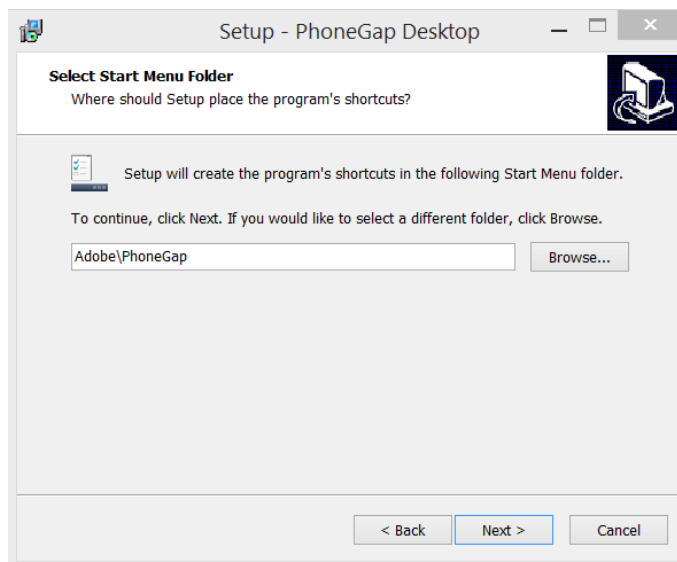
KUVA 42. PhoneGap Desktop käyttöehdot

Seuraavassa ruudussa ehdotetaan valikkoa johon ohjelmisto asennetaan. Hyväksy ehdotettu hakemistopolku painamalla Next (kuva 43).



KUVA 43. PhoneGap Desktop asennuspolku

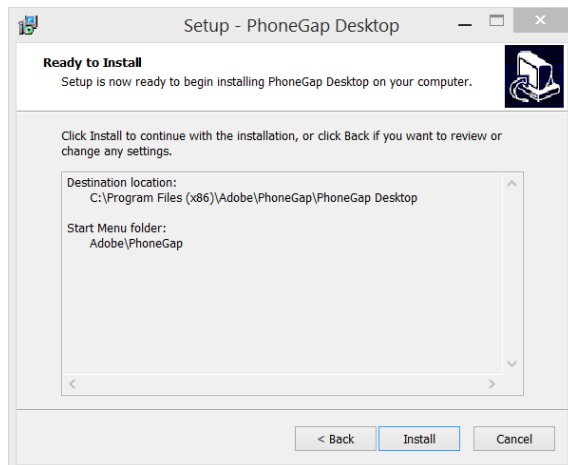
Asennuspolun määrittämisen jälkeen asennusohjelma kysyy, mihin Start Menun hakemistoon ohjelman oikopolut sijoitetaan. Mikäli et halua määrittää omaa hakemistoa, valitse suoraan Next (kuva 44).



KUVA 44. PhoneGap Desktop oikopolkujen sijainnin määrittäminen

23(24)

Tämän jälkeen asennusohjelma on valmis asentamaan PhoneGap Desktopin tietokoneellesi. Mikäli olet tyytyväinen antamiisi vaihtoehtoihin, valitse Install (kuva 45).



KUVA 45. Asennuksen aloitusruutu

Kun ohjelmisto on onnistuneesti asennettu koneelle, asennusohjelma tuo näytölle ruudun jossa tämä asia kerrotaan. Viimeistelläksesi asennuksen paina Finish (kuva 46).



KUVA 46. Asennuksen viimeistelyruutu

Kun PhoneGap Desktop -ohjelmisto on asennettu koneelle, täytyy vielä puhelimiin, joilla PhoneGapilla tehtävää ohjelmaa aiotaan testata, asentaa PhoneGap Developer -sovellus. Kunkin alustan omalla kauppapaikalla on kyseinen sovellus ilmaiseksi ladattavissa. Asenna kyseinen ohjelma puhelimeesi tavanomaisesti ja aloita ohjelmien käyttö esimerkiksi www-sivuilta <http://docs.phonegap.com/getting-started/3-create-your-app/desktop/> löytyvän ohjeen mukaisesti.

**Yhteenveto**

Tässä käyttöönotto-opasteessa käytiin läpi ohjelmistot (pois lukien Java), joiden täytyy olla asennettuna, jotta PhoneGap ohjelmiston saa toimimaan. Ohjelmiston varsinaisesta käytöstä löytyy kattavia opasteita internetistä, joten niiden kirjoittamista en nähnyt tarpeellisena tähän opasteeseen.

Mikäli tässä käyttöönotto-opasteessa olleet muut kuin PhoneGap ohjelmistot olivat asennettuina koneella valmiiksi, oli PhoneGapin asentaminen ja käyttöönotto nopea toimenpide.

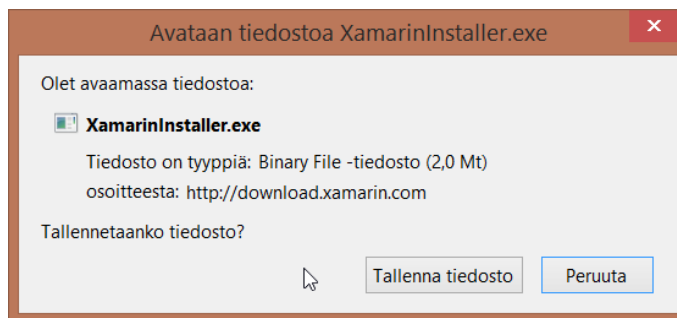
**Liite 4. Xamarin Käyttöönotto-ohjeistus**

## Xamarin käyttöönotto

Xamarin on maksullinen ohjelma, joten voidaksesi käyttää sitä täytyy tuote ostaa Xamarinin verkkokaupasta. Xamarinin verkkokaupan osoite on <https://store.xamarin.com/>. Valitse haluamasi tuote ja seuraa verkkosivujen ohjeita.

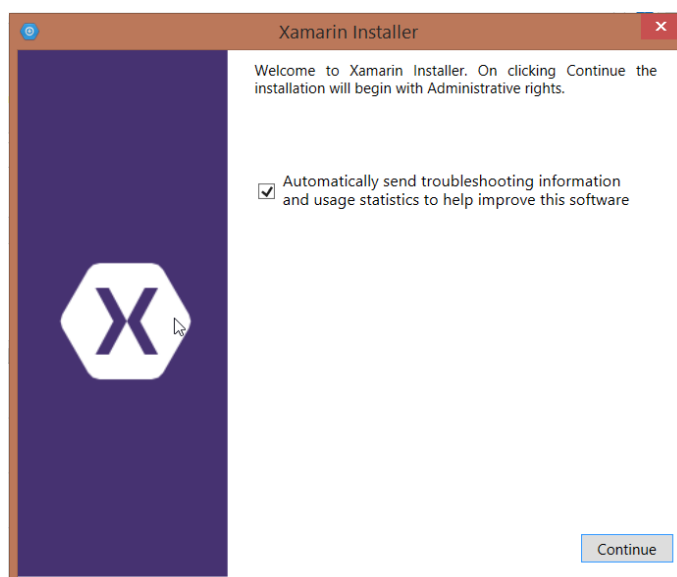
Kun Xamarinille on laatinut tilin, voi omia tietojaan hallita osoitteessa <https://store.xamarin.com/account/my/subscription>, ja mikäli et ole vielä ladannut Xamarinin asennustiedostoa, voit tehdä sen osoitteesta <https://store.xamarin.com/account/my/subscription/downloads>.

Tallenna lataamasi XamarinInstaller.exe-tiedosto valitsemalla Tallenna tiedosto (kuva 1).



KUVA 1. XamarinInstaller.exe tallennus

Paikanna lataamasi tiedosto ladatuista tiedostoista ja käynnistä se. Xamarin Installer käynnistyy. Paina Continue (kuva 2).



KUVA 2. Xamarin Installer tervetulo-ruutu 1



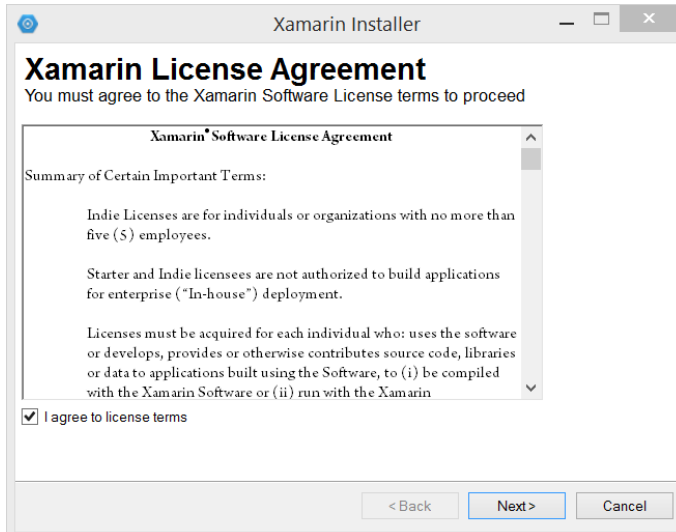
2(6)

Seuraavassa ruudussa asennusohjelma kertoo asentavansa kaikki Xamarinin tarvitsemat ohjelmistot. Paina Next (kuva 3).



KUVA 3. Xamarin Installer tervetulo-ruutu 2

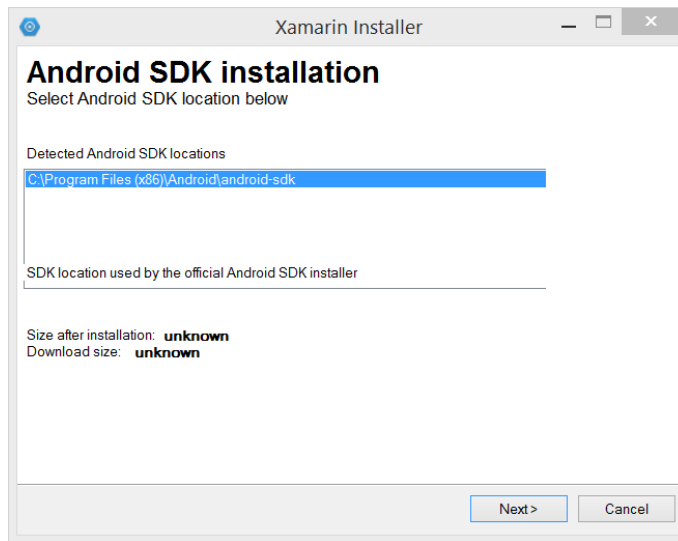
Seuraavassa ruudussa näytetään Xamarinin lisenssi. Jatkaaksesi asennusta hyväksy lisenssi ja paina Next (kuva 4).



KUVA 4. Xamarinin lisenssi

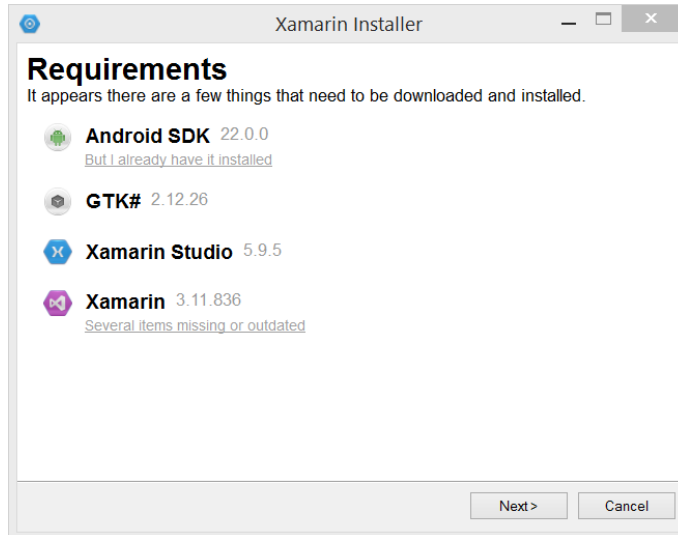
3(6)

Seuraavaksi asennusohjelma tarkastaa Android SDK:n olemassaolon koneella. Paina Next (kuva 5).



KUVA 5. Android SDK asennuksen tarkastus

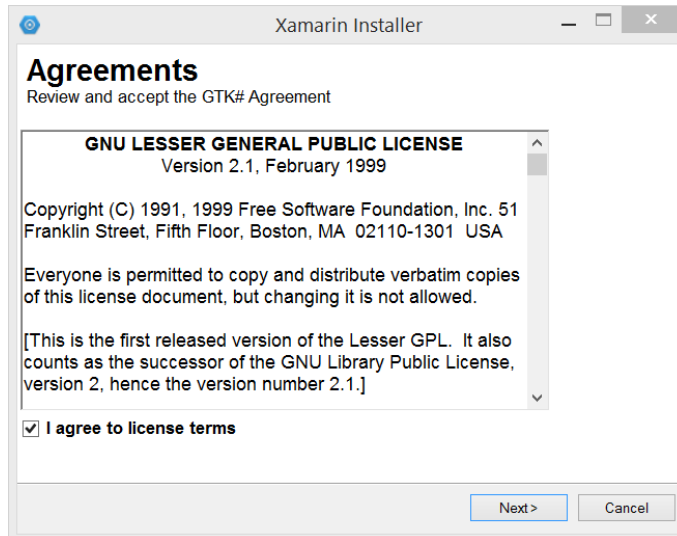
Seuraavaksi asennusohjelma ilmoittaa vaatimuksia: mitä osia täytyy ladata ja asentaa, että ohjelmisto toimii. Paina Next (kuva 6).



KUVA 6. Vaatimusruutu

4(6)

Seuraavaksi asennusohjelma pyytää sinua hyväksymään GTK# lisenssin. Hyväksy lisenssi ja paina Next (kuva 7).



KUVA 7. GTK# lisenssin hyväksyminen

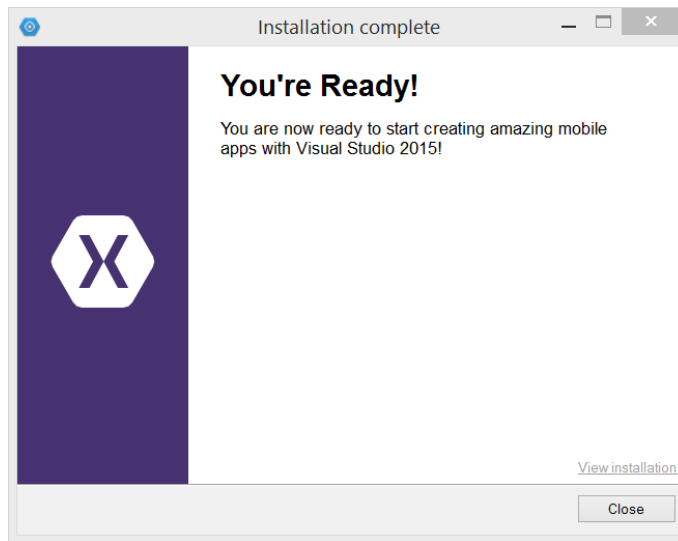
Asennusohjelma aloittaa ohjelmistojen lataamisen ja asentamisen. Odottele rauhassa kunnes asennus on valmis (kuva 8).



KUVA 8. Ohjelmistojen latauksen ja asennuksen edistyminen

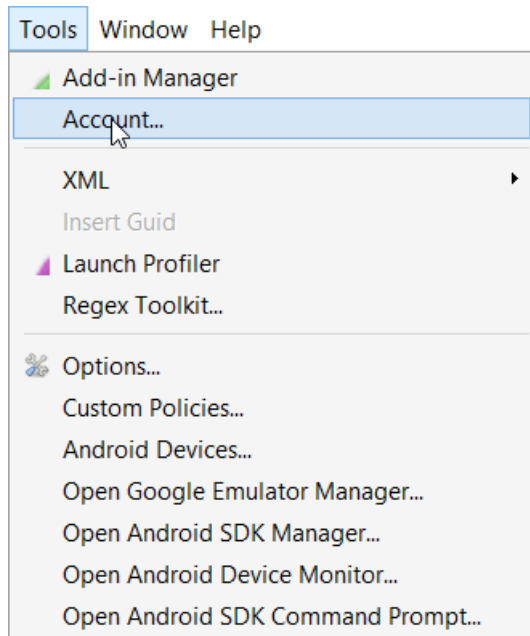
5(6)

Kun ohjelmistot on asennettu, tulee ilmoitus asennuksen valmistumisesta. Paina Close ja käynnistä Xamarin Studio (kuva 9).



KUVA 9. Ilmoitusruutu asennuksen valmistumisesta

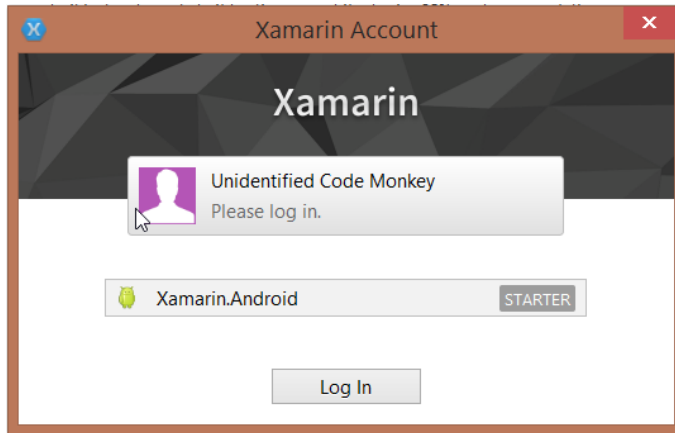
Xamarinissa etsi Tools-valikko ja sieltä Account-alavalikko (kuva 10).



KUVA10. Tools ja Account valikko

6(6)

Kirjaudu sisään Xamarinin tilille painamalla Log In ja antamalla seuraavassa ruudussa sähköpostiosoitteesi ja salasanasasi, jolla olet tilannut Xamarin subscriptionin (kuva 11).



KUVA 11. Xamarin Log In ruutu

Kirjaututtuasi ohjelmisto ilmoittaa tilauksesi muuttuneen aktiiviseksi ja kertoo, millaisen tilauksen olet tehnyt. Olet valmis aloittamaan ohjelmoinnin Xamarin Studiolla (kuva 12).



KUVA 12. Xamarinin aktivoinnin näyttävä ruutu