<meta name="viewport" content="width=device-width; initial-scale=1.0; "\>Event ID: 1922710
Event Started: 4/25/2012 2:00:00 PM

.
>> [Please stand by for realtime captions clack clack [Please stand by for realtime captions]

>>> good morning everybody. This is Darryl diamond with DigitalGov University at gsa. A quick technical announcement. If you are having difficulty connect together webinar give us a call at 202-208-0668. Or you can e he-mail us DigitalGovu at gsa.goive. We will be starting in a moment.
>> Good morning this is Darryl diamond with DigitalGov University. We are happy you can join us for the webinar on the micro-- windows plast form part of the mobile development series. Before I start the talk by introducing the Jacob percell all attendee lines are muted and if you have a question, you can go ahead and type it into the chat box and we will do our best to get to as many questions today as time permits. So I want to start by introducing Jacob Parcells of our mobile team.
>> thank you very much. Good morning everyone. Thank you for joining us on this mobile platform webinar today. We are having -- this is a number of series we are doing. Last time we were doing blackberry research in motion today windows and with us Joel Reyes who works for windows and he is a microsoft developer and works in the platform evangelism division. And what Joel is going to work us through is talk about how you might develop a windows phone product and how you would work on getting it on to the windows phone platform and other things. So without fur further aday here's Joel and as Darryl said ask your questions via the chat box and we will ask at end of the presentation. Thanks very much, Joel, take it away.
>> thank you Jacob. I want to thank everyone for attending and thank the organizers for extending an inhave I toyings to us to present the windows phone and. What I want to talk about is what we are going to discuss today. So fundamentally I would like to cover three aspects of our platform. First I want to share the rational for why the phone is different. Doing a review of the architect and platform capabilities, again, indulge ourselves for a couple seconds on the user and what does it mean and why we choose to go in that route. It's important coming to designing your application. In adiseur and data market. I will try to cover it in the first half of our talk and the second half go through the mechanics of how you build an application and how you publish it et cetera et cetera and in the context of all that, be able to answer your questions. So we will be be monitoring the view at that time. Also, throughout the talk, I would like to make sure I cover this items that we share how to prepare for creating an app in the marketplace what is the creation process many application. Troubleshooting tips and tricks available. Guidance and how you go about managing updates and maintenance and any other input on how to help you guys to do services in our platform. So be this will be more clear when we go through the actual demonstration.

>>> also you will see the infomercials popping up. This is very important. This is resource we have made available to the community at large on things that you can tap into to actually get you started. I put together the info kit and this is the web address so write it down and I will make it power point available so it starts right there and that will give you a collection of resource that is I put together, tools and training references and data references so you can get started quick. Things like the training kit for instance which is the full hands on experience for developers to get started, there's videos et cetera, et cetera. So, keep an eye on those things as they he come up. All right. So, the one key thing that microsoft took a step back of the mobile market and the solutions that were available given our experience with windows mobile and all that. And we said okay, what is the one thing that we want to make very strong in windows phone and it came down to this notion of putting people first. And what that is philosophical driver for how we went about designing the window experience and what it means for consumers of the devices. And I think the animation will kind of help visualize this. When we look at the market, even today, you know, there hasn't been much progress in the user experience. We have throughout all the different brands a grid of icons that allow

you to launch an app. So, that works fine but we took it a step further and says this is my device and should be about my things and put me first and put the people I care about first. So we took a different approach to that and came up with this commitment in the user experience that says, the device itself brings to the surface the things you care about. You know, your people, meaning through the -- do the people hub I can say not only my personal contacts but for my corporate system, from any third party e-mail system like google and so on, linkedN, and Facebook and it allows me to bring to the surface the people I care about. If I have contacts I want to keep in touch with and I want to know what they said in a given social cycle circle on the way of Facebook and so on. I want to see that. The things I care about my pictures of my kids and family and whatnot. So, it's important to know that the first release of the device was heavily focused on the consumer experience. We have a lot of capabilities for the enterprise e-mail and corporate e-mail and contact and calendar, et cetera et cetera. That's the fundamental difference and the drive for the rest of the experience as far as the user experience throughout the whole device. And obviously, we have made it possible for developers to also you know, adopt the same philosophy as they build their own applications. So, from this point on we are going to basically concentrate on the fundmentals of the platform. We are going to share with you you know, what are the foundational elements for the hardware and software and the application layer and what are the cloud and services that have been integrated into the experience both from a consumer point of view and a developer point of view to be able to leverage the services to build your own solutions. So at the foundation level woof the hardware. This is where you have things like sensors and cameras and motion on the chill the spec for the chasse and all that. That's fundamental stuff we have basically, I hate the word dictated but we dictate what the things are and the reason being is the windows mobile world, they had a lot of flexibility on how they would use the software to build their hardware and we stepped back and said no we need to take a better control over some of the assets so we don't end up with a endless numbers of things that makes it difficult for both consumer making a choice of what device brought us all as developers because they have to count for too many variations and we wrote it down specifically and things like the buttons on the device. So every device no matter who makes it will have three buttons and a proms. The home but the tonight windows flag and a search button and a back button. It will have a up and down, power button and camera. That's it. No more buttons. No matter who makes the device. Those kind of things so we can can have a predictable experience from the user and developer in the latest of the platform we added capabilities for performance enhancement we have inherented from generational driver selection from the defendants to being and added secret service ce so if you need to store data locally you can do that. And we have allowed the ability for you to build silver light and we will dive into it later. We added fas app switch which means if I use a application and switch to another app and want to go to the prepare app we have capabilities so the application happens instantaneously instead of launching it from scratch giving developers to take actions like storing data you need for later, and the application is back in the forefront and you can tap into that and create experience of the user without starting from the beginning which happens otherwise in other platforms. In fact in our own platform when we first came out. We have background agents that allows you to perform something from the background even if the app is not running and this is the manage approach so we don't have to have application running in the background consumer memory and resources et cetera. I mentioned integration between the two and we will dive into that later. We have the notion of app connect that you do a search for books. And on the global Bing search. If you build an application you want as a result of a search, the app connect allows the capability so your application will be aware of search results that have been the users can switch over to the application that it has Domain capabilities for the context of that search I did. It has push and alerts notification and do not allow access to the calendar and contacts so you can build a app a business card and so those are adissability capabilities that is the latest release up until now. And on top of that you have services. So, that's a summary of capabilities additions we have done past the first release of windows phone. Beyond that, you get the emulateor, again this is not a simulator an emulateor. So it's a buy nary image for you to test the apps and they will run as if it was on the device itself. We have added profiler so you can fine-tune the application et cetera and you can build apps in c sharp and vb.net. So vb developers can build the apps and I am not sure if this is relevant but we have an add to allow developers to build app and provide them for free and provide apps within the apps so they can generate revenue that way. The development environment is advicual studio if you don't have visual studio 2010 the professional version, when you down load it it will detect it and automatically give you access to the visual and it installs that and then expression blend is available free. As well as xna game studio. Everything you need to build windows phones solution is free. If you want to tap into enterprise level capabilities with visual studio and have a full life cycle management that's available through the commercial version of visual studio. But fundamentally if you want to get started with this and want to stay in a couple hours at night at home with it, everything you need is available on the sdk and you down load it. Beyond that you have portal the app portal where you submit apps for certification and gives you access to the resource that is are available to the msdn services. So, here's an example of the hardware foundation I was talking about. When we started we decided it's going to have this resolution, 800 by 480 the pixal and this characteristics for capacity touch and sensors. If you want to provide a camera it has to be 5 mega pixal at least and no less. Those things so baseline of capability that are predictable across the different devices regardless of the. And then we have another support for one more processor. We made available the camera and sensors and things you can tap into directly from an api point of view within the app so if you want to have the camera interact or application interact with the camera you can tap into the camera stream from the app instead of the camera application that comes with the device it sit cement it allows for -- itself. It alaws for better solution more integrated. And these are the elements of the platform. We have the scream and cloud. On the screen you have the ability to target windows phone for building apps, x box for building games and windows 7 building the normal apps. So beyond that, you have the ability to within the phone aspect of it, to rely on a manage infrastructure part of the framework and it has all the capabilities we provide as part as manage and the access to all the sensors and location information and notification which part of the api. On the tool side you have visual studio and expression blend that gives you access to the emulateor and guidance and you get communities if you have questions you can ask there and people from the windows phone team will answer that so you getdirect answer from them or from the community at large et cetera. And that's beer you -- that's where you build the application package it apply verification to it before you submit it to the marketplace and we will see that as we progress into the demo. On the cloud side support for windows azure. You can use it as to connect to other systems or you can build services that you actually -- so you have a full api that you can bring down to the desk top to build and test the applications and once you are ready for performing you punish it -- push it up to the cloud and within the windows phone app reference that and consume the services using standard protocols. When the portal you need it for registration and that's where you submit your app for validation and they will provide -- it will keep you up-to-date on the certification process and the publishing mechanism and so on. That's kind of a fundamental elements of the application management beyond building and testing. So, these are the stack of the platform. You have a hardware foundation and then we have colonel level things like security management and very strong security mechanisms for isolation and so on. And work capability and storage and interaction with the storage and all that is all that stuff is at the lower level. And you know anything to do with the hardware and the integration with sensors its pardon of the colonel as well. On the application layer you have app management that handles, you know, installation and creation of the sound box for execution and that actually you know is response to the capabilities of the app and getting into detail about that later. You know, the licensing for the application, et cetera all the stuff is transparent to the developer and the model takes care of that. The ui model we rely on a shell frame where we compose the ui basically. And the way you navigate it through the pages, elements is through session manager and this works similar to where you have pages and navigate the pages and in essence the referencing of the pages happens in the urifashion. The vendering of the elements happens is done by directX. So direct 3D does rendering and leverages the gpu chip built into the device so you get amazing performance. I mean, it's just incredible how performance this is. It's a element of competition of the elements on the shell frame are created so that you can compose things and allows for integration with some of the what I am trying to say motion elements of the uri. And cloud integration we have x box and integration into the x box and location based and push notification handled through the cloud and so that windows live id there's not an api for it that is intergrated into the tool set but there are ways to actually leverage it for things like authentication through live id and I am running a series a secure line application for windows and if you go to the blog, ms blogin to msdn forward/Joel you will find my series and that should give you a good set of information how to build secure apps and how to authenticate against what provider you may have and leveraging identity management et cetera. On that you have the clr. And that level up the frameworks. So you have three times of silver light, and what I am saying. Xml -- tm -- xna and the ability to build apps that are mobile web apps and java script. The Joyce it changes -- it Joyce it changes everything. It is leveraging built into the device to access your web mobile app and go from there. Your applications on top of all that. Okay. So the things you have to worry about is simply the business logic and application logic and ui design, which ever choice you make of ui experience. Everything else is there on the device or accessible through the api. The key things is if you are a developer you know I would say you know 85% of what you need to build windows phone apps. That's why building this applications on our platform the developers can do it so quick because they know 85% and it's a matter of learning about the design experience and some of the elements of our Metro philosophy et cetera, et cetera. So, that's the good news is that as you tap into the platform you go on in knowing most of what you need and all the tools available for free, I mean, that's pretty amazing. So, I can't think you would be building games but gaming is prominence beyond just entertainment so it's been used for education and other purpose so maybe there is an opportunity, but I think for the most part silver light is what you want and it's event driven ui framework and you can build applications quickly because you can dart and drop the controls that are built into the platform or third party controls that you may purchase. It's up to you. And the demo will show a little of that. So this is now we are in silver light 4. So these are some of the enhancements. I will highlight a couple sockets so if you need to build a application with high performance connecttivity sockets is is a way to go. We have the ability for you know, get text from the clipboard. For some Asian companies imes so if you need to have them more constrained experience you can allow for that way. And web browser is based on latest release of interneexplorer 9 so there's some other things like I mentioned it's transparent to the developer. But it's good to know because it increases the performance of your application. So what things can you do? You can chews choose to built a silver light or xna app or combine the two and you can choose c sharp or play and manage media, you know you can access the phone book and messaging infrastructure and leverage things on the clouds like push notification so if you've service on the cloud and want to send out notifications to the users, through the app, so the app can register we have service on the cloud and it receives notifications and we have a simple api layer to allow you to get receive the messages. And full service for location based is predominant nowadays. Not sure if selling the apps is relevant at this point. But I am starting to see very early trends around this saying I am government agency I am going to publish an app but we are going to require 99 cents or whatnot for the apps to cover cost. So there's a possibility for that to be an option and it's kind of starting to become an option for some customers. But, again, given the being a government agency you know, that's kind of up to you how you want to pursue that. This is the most powerful slide I think in terms of the depicting the capabilities of the platform in the current form as part of the release. Camera capabilities and access to the pipeline directly. All the sensors are available to you. All the working capabilities being a stows data through which you can encrypt or elements of the data base. Phone contact, you can read and write to that phone calendar you can read. Some of the launch this is important because these are capabilities built into device experience that are available to you as a developer. Say you want to trigger a text message from your app or within your app. You can tap into the text message interface and within your application using the launches and chooseers and provide the information to prepopulate the information and then use the interact and choose the target first who they want to send it and send the message. Likewise with e-mail but that capability is built in so you don't have to recreate a wheel. Leverage it to the launches and chooseers to do that. And controls. We have controls that are out of the box and controls you can build or controls that you can down load from open source places like complex or controls you can buy from third parties. So all the possibilities there are covered. You shouldn't have any problems finding the right control you may need for your app. The place where you go to start building apps today is aka.ms/getphone giving you access to the sdk and it will take care of itself and put everything you need and when it is done launch and start going at it. That in combination with the info kit should give you everything you need. When you talk about what do I need to prepare creating apps that's gives you access to everything conceivably you may need to get started. Okay. This gets you going to the cloud many if you need to experiment with the cloud, you go to aka.ms/getazure and that's gets you the sdk down load and you have the option to allow you access it for free for a period of time so you can get a taste of the app before you make a commitment. So in terms of user experience I want to very quickly mention this. If you look at this picture here, we basically have made a commitment as far as the design that we were going to go with an approach that provides the notion of bringing data to the servers allowing the user to interbeing a with the data and get away from chrome and the official elements that mask the user appearance to the real world. We tant to remain authenticly digital so we can leverage the experience. If you look at different pictures, we pretty much observed this in different experiences. When you go to the airport for instance or to the bus stop or grocery or things of that nature. They basically remain true to that particular experience and bring data to the surface you know elements of fun and space and -- font and space and stylish element of it color you know, good balance of color et cetera. But it's really a simple experience that is powerful allows for things like this. This is zoom so I have the ability to interact with the data. So I have that experience right on the interface. If you look up -- if you go around when I traveled sometimes, I noticed things like this. And I take pictures of it. So if here's an example of what we are talking about. If I just look and got off the plane I need to know where the backage claim is and ground transportation. Very simple and straightforward and basically if I don't know how to read I can know and tell because it gives you simple iconic elements to point me in the right direction. Likewise here. So when you use our device it's similar to the images you are looking at and that's the whole knows of Metro is keep it simple and four -- power tulful and -- powerful and bring it to the surface. Here an example what the experience looks like. So you get the main screen with the tile system that's connected to the notification services to let you know how many e-mails you haven't read. Contact information is straightforward. Data and picture of the individual. Et cetera, et cetera. If I have messages I can navigate and interact with the data. And here's the example of the software on the device for playing music. It's all data driven and no chrome and the data you need presented in a way that's beautiful but functional. So what we are talking about here is principles. Let's keep it light and simple. Let's leverage the typography as much as we can and add motion so the user knows when things are happening. And let's concentrate on content itself where there is a media content or text content and let's walk away from the chrome element of the ui design. And let's keep it honest in the sense that is digital experience and night real world we will don't need to represent this experience with iconic features of the real world because it doesn't add much value if any at all. Okay. So, data. As far as data and there's a question what else you think will be -- will help service in the platform. I would say data is where it starts. You know, we have windows azure that allows you to build core the services and consume the service and they are deployed throughout the world. It's been announced the north America region have exten or expand to cover more regions and it's not reflected here. You get to choose where you want to store the data to put it closer to the user if you will. These are some of the capabilities of that platform providing. I am not going to go into all the details. But what I want to mention is the windows azure trial that aclass you to get an account for free for a limited period of time many these are the features. 750 hours of access and 20 gig for storage and 20 data transfer and you get to experience the whole azure technology without any commitment up front and then, you know, based on your experience you will make a choice. That gives you everything you need to actually experiment building windows phone applications that consume trial services at least adiseur trial -- azure trial service we have a architecture odata then we have market -- data market which allows basically the access to all the data that's hosted within our windows azure system where there's storage and third product cloud service. You get the ability to go and discover what is available, what's free and what's not free, et cetera and we will get to see that in a minute. Okay. There's also other services out there like data.gov has its own data services that are accessible. You can see some of that reflected when I produced that but probably now more than that. They have publish the data on the marketplace. Data marketplace. So you can choose to either, you know, have your own data services, internally publically but internally or you know you can actually leverage-- delegate it to something like data services -- data market phone when it's azure and developers will browse it and play with it and decide what they want. And I will throw that through the demo. So in doing that process I will show you a little bit about this. The azure data marketplace iscollection of data both free and pay per use and worldwide and accessible and the best place to have data. You can have data so the developers can go in and build apps. All right. The marketplace. Quickly I want to get into the demo. You can register as a member as an individual or a corporation. It's $99 a year for students weprovide free options. Once you right the application we provide the service for submission of the app and give you what did not pass because we make sure every test pass before the application comes to be be published and that's good for you and the users and for us because we have very strong checks in there of security and otherwise to make sure the application will behave. Again, everything is available at this address And also I want to mention things like the web. We offer the web matrix. You can down load that for free. It has nothing to do with the phone but we advertise it because it's something we like to measure as well. All right. So let's go to demos and q and a. How we doing in time here? Perfect. Perfect timing. All right. So, the first thing I want to do.

>> You are in good shape.

>> I want to address the question how to prepare an app. I gave a synopsis give you -- giving you the information about the platform and services and so on. Basically the way this goes is like this. If you are already a windows developer or if you are not either way but particularly if you are, the experience for building an application is exactly the same. Okay. You go into here and say I want to build a new solution or open one that exist. You know you are going to get to choose from a collection of templets so notice that I have choices here. So in my case, I opened to c sharp and I could have installed db as well but my defought was c sharp and there are other choices in here you notice that you have different choices for what types of applications you can build with that. And notice I have silver light for windows phone and it allows me the types of application and I can build an application to start with a page and I can start with a templet and if you have data collection and want it binded to a list and make that your start off point for the application we can provide you with a templet that provides the framework or it can be a class library you will use in another app et cetera. There's the Panorama application. We use it a default for many of the apps instead of the explicit pages where you have a collection of buttons which takes you to a different page or back button within the app. There's this notion of Panorama. Panorama think of it as all extensions of a uri that goes beyond the screen and you have the window into that. Within that by swiping left to right you navigate from page to page but it's built into the experience. You don't have to account for back buttons because the device has the back buttons so when you navigate from one to another you can press back and it will take you back or swipe in this case. I will use this as an example to give you a little sense of the mechanics of building a app. And then I will switch to one that's built to show you how you integrate your applications into consumer services from the cloud. Notice I have two option the original version of the os and the update that was provided. I will going to choose that. Because I chose c sharp everything is going to be c sharp. You could possibly build a class library in vb and consume it from here because you are deeing with the code so it's the language that is relevant. So you have the flexibility how you want to do that. If you notice, the solution is the same as the silver light application or anyone of the other types of apps we can build. What you see is what you get ui experience for your ui design. You get a collection of controls that are, you know, built into the extension for the studio and part of the windows phone sdk and you get a page and you get a code behind so what you see on the screen is defined. Whether I change it or it has basically by direction. I can say so let's say I do -- notice how I go over to the control here, we use expression or whatnot same kind of thing. Okay. So I can possibly take this application and edit it with expression blend see that. I can take this application and go to expression blend and edit an and the system will

notice the change and say do you want them and yes and it brings it back. It allows the designer to restart the application design in the pending of the development or the developer in the pending of the designer. But they get to interact with each other's work without stepping on each other's foot and that's the concern because it's power because if you have people with specific skills around design they can go do the work and you can do that into the development work et cetera et cetera. So it keeps it separate. When you compile in am cation you get the same experience you would get any other type of solution right? So once you compile you can run the app and bring up the emulateor. And in this case I put a Nokia skin so you can change the skin of the emulateor until your heart is content. Those are external skins and what you have is an exact replica of the device in your hands -- as if it was in your hands. Once you runt apps it will deploy the application to the emulateor and allow you to run it. And in this case it's Panorama app 1 and I didn't chack the -- change -- change the name or anything like that many it has a screen and you can interact with it. I have the list here that I can bind to some data somewhere. I have a list that allows for pictures and you can have picture and name and subtitle sort of thing. And just like the plane picture list that's name and subtitle. You can put title and subtitle. That's as far as the Panorama. So this is just a templet app that you use to get started. From there, you can app pages and so on. So -- add pages and so on. So let me close this here. And show you that adding a page you go to the application and say add a new item and when you add a new item like any other experience, I am going to add a landscape a user control whatever it is I want so I can add a new page to my experience and here then you can go ahead and design this. I am not going to go through every detail but let's say you want to add buttons. Drag and drop the buttons on the screen. Maybe you want to add a text box. You can add that and put it wherever you want and name it whatever you want. And you can add you know, event handler. So let's see what's going on and you notice I select the text and it was selected down here and gives specifics about the type of the horizontal alignment the margins and what's the name and text box. So I can say text goes here. Okay. And so on. So, basically you have a correspondence between the uai and experience. From that point on I am going to -- I used the wrong -- put it in the wrong place. Where is it now? I will fix that this way. So, I want to -- let's say this is your application as you wanted it. It's compiled and everything so if we look the the property of the product,. [No audio]

>> you see that -- are you seeing any kind of student project. Here's your images for the application and you have the background and you have the Panorama view et cetera. Notice I have here a the control collection and it has the Panorama api taking the stuff created in the application and compiles it into the dll just like the app in the application. We have a manifest which I want to open up here. It shows you basically the elements that are important to the application like that's where you got create a list of the external resource available to the app and to the dll. And you have the manifest. The manifest is important for many reasons. This is where you supply the capabilities of your app. This information here is taken by the marketplace certification process to actually disclose to on the way of you know what are the capabilities you have that it relies on. It's also used to influence the environment it's created, the custom sandbox that is created for your application specifically. It's based on this capabilities. That means is that if I were to remove the important capabilities from here, right and within my application I want it to access the web, it will tell me that's not part of your app and that's a good security measure because it prevents situations where you may not disclose that a use doesn't work and try to use the code that you build to tap into secret websites or can send personal information or whatnot. All that is impossible because of this ability we have to enforce the capability. You have the camera and browser and any sensor. You name it, any capability we mentioned up front when we started the presentation, are expressed here and it's your -- you as a developer have the obligation to disclose the things and your application will only give access to those things. That's very important. Then what you see here also I see a unit of deployment okay. This is where you submit to the marketplace. If you look at what's in there we will rename this thing so we can look inside of it right. I am going to call it .zip to package it to fast forward it to the marketplace and the server will take it and crack it open and take the pieces out and you know, apply all the rules. And then from there, when the application gets published, when you go to the device and browse the marketplace, and you say I am going to install this app, this file is

transferred to the device. Os on the device will do the same and crack that open and pull the pieces out and apply all the rules that it needs to, you know, provide the certificate to the app noter is testify kit but rather the isolation chamber, the time box and it provides a what you call it, when you basically when you buy a piece of of software and gets a license. It applies a license spie sick to the application. This is the extreme we took. When that -- spie civic the extreme we took. When you take it from one you can't install it on 2 because it's specific on device. There's no change of somebody hacking into the device and doing weird shengs but notice I have now -- things like that. But notice what I have in there is again the icons you need for the application, you have the manifest, you have the two controls you need and the application manifest and the deployment manifest. So, that's what basically is inside that file. And that's what basically is transferred back and forth up and down the marketplace and into your device. That's an end to end view of what happens in terms of building apps and debugging it et cetera. Beyond that, and let me save sometime so we are good on timing. Yes. Just quickly, to show you that you can actually you have full control. So if I worked around this application it's going to stop here right? Because I put a stop there to allow you to step by step debug the application. So all of the debugging camability are available to you just like any other -- casability are available -- can be capabilities are available to you just like any other project. This is really important open market test kit. It -- you build your app and say okay doirnghts want to have to keep going back and forth and submitting the app certification and applying the rules and tell me what didn't pass and I come back and fix it and submit it again. This tool basically says let's do that locally. So it applies the rules locally in the project so you can check the potential thanks could go wrong. And then once you do all that, you submit to marketplace the cangs are going to be minimum you have to address issues. And it cuts a lot of the back and forth and it canly things like -- it canly -- particularly you have up to 100 free am cages you can submit in the marketplace including updates okay. So if you are building an app you don't want to a uploads just on updates. This will catch them and cut that out so you can do it locally. So that tool where you have application details and you can provide all the images that you need when you browse the marketplace. It will show you pictures and the small icon et cetera, et cetera. It has automated test so you can go through the test and say what xap package requirements and capability validation and monitor test for launch time so you can launch the app and start manipulating the app and the system will track metrics that are summarized to gib you an idea for memory consumption and use of the back button et cetera. This is very useful in terms of minimizing interaction with policy and testing. That addresss guidance and microsoft and windows phone app maintenance. You submit the application and an update, the update process is handled for you automatically. Marketplace is going to have changes there, device it will send notification to the device that the application has a new update. The os will handle that life cycle of updating the update and installing it et cetera, et cetera. You don't have to do anything other than submit the update to the marketplace. Okay. All right. So now I have ten minutes so I want to show you another application. I am going to spend five minutes showing you the mark place and then five minutes -- so let's do data market. Okay. Data market is where you go to discover data right? The first thing you have to do is log on using your live id. So you go here. Okay. Once you are there, you have an account. You can go to my account and you can check basic stuff. And you have primary account and customer id and check whatever apps if any you may be using. And in this case, I am consume eking data from my data so data -- consuming data from my data so I can explore what he available and gives my the -- what's available and gives me the ability to explore the sheet in excell and it's free and so on. Beyond that you have a account keys for the user to use the data and you have to provide this key. Tough use the user id and this key associated specifically with the data source. Those things combined from your application will give access to the data source to the user. Let mow quickly go in here again because I want to show you some of the cape abilities that are -- capabilities that are here. Application, show content. All right. So when I go here, okay. Notice that I have the ability to create queerys -- Quarrys -- querys to get the data source. And in this case they provided the full name of the state. The city I will -- all right. So here's crime data for north Carolina right. In the cities there's the cities and year and population and what crimes have been committed et cetera. Also known up here you

have a url for the current expressed query and a service root url and that's the entry point to the data source from the app. So, you can down load this data into excell and look at charts and look at the version of this if you want to do that. You can vishual or export it. This is the actual -- visual or export it. This is that will filter it down to Carolina. You can express that or you can do it through your application or url. Let's go to the application itself and show you how you go about consuming the data from the windows phone app. The first thing you need is a service reference. Service reference is basically a service reference to a end point that prointerface or service. From there, the system will extract the data from the service and it will do the magic behind the scenes. You don't need to worry about it. If you don't know what service development you know about that. Then within the application you create a uri which is the end point and exactly the same end point I showed earlier. So, it's going to provide some kind of identity so it can validate, you know, access to that data source information. There's the key and all that. And then, beyond that, it's going to create a context and create a data service based on the context and now here, I can actually rely on link to sequel to express my query. All right. And it's the same query we saw earlier in the uri form being done directly through c sharp or whatever you choose and I have access of the level access to all of the properties so I made a collection and I can reference it by name and provide values and I can specify things like order by, select, et cetera. Anything supported by links sequel is supported here. And now a que he ry collection that I can -- query collection that I can put in the system and bind to my uri in any fashion I deem necessary. That's how you go about leveraging that data from the web and the rest is basically good old fashioned development about you know you have data series. And it's consuming a control in this case a custom control from component one. So if you look at that, I am not going to dive into it but I have three reference for three types of controls that are third party controls registered into my system. Now I can reference though and have access to those controls. Okay. So if I were to run this, again, just going to load it to the emulateor. Create a sandbox based on the capabilities that have been specified and launch the ui. So there it is. Okay. So now I launch it and I will interact with it. In this case the application has a list so I am binding this list to the actual results. And I have the charts taking the same data and populating the chart with what you see behind in the code and then a visual experience that's more telling if you will than just the raw data. And here's a trend type of chart. This shows you the interaction between the windows phone application that's consuming data from the cloud using standard protocol and leveraging data and third party control to create a better user experience. Okay. So I think that, you know, in a nutshell those. [Audio not understandable]
>> the mechanics of building the app and deploying the app you know and so. [No audio]
>> and the ability is then to see your application. So I am going to stop in few minutes. Again you need to log on as for an account and then you provide a reference to the file and bring the file in and apply the reference to images and things of that nature and brings it all in. And it is going to trigger the verification process and take couple few days, couple three or four days to certify the app and they will note fa fie you back and if it doesn't fail they will say it's ready to go and you can prespecify or don't publish or publish it. You can publish apps privately but I only do it to my own group of people within my enterprise and you can submit it as private and it will give you a url and you can disseminate it privately in a secure like channel e-mail so only the people you intend to have access to it will have access. And overlay on top of that andauthentication mechanism that can leverage identity management out there to secure the application. So that is a second -- two layers of security there. So, that allows for you to have an application and marketplace that's not visible to the world and deploy it to your own constituents. Okay. Let's switch over to questions now because we have two minutes left. Any questions.
>> Thank you very much. I have time for one question I want to ask very, very detailed description of the windows phonedevelopment process today, does windows do anything on the back end when I submit an app? Is there anything that I should expect to hear from them or does it just publish once I have submitted the app.
>> there's a full life cycle that takes you through the process and keeps you notified of every point. Submitted it, it's been tested, and it got tested and

failed and this is why it failed and gives you a report specifically telling you where it failed and why. If it passes then we will tell you it past certifycation and ready for publishing. You get to say if you want it published or want to be notified. When you go to the portal you will have full control over whether or not the application has been -- let me go in there quick and show you this because I have an app that I submitted that failed in certification. So let me log onto this thing quickly. And.
>> because we have to move along quickly if you can show us that, that would be great and then we can wrap up. But we do appreciate your time today and the expertise you've shown us.
>> Absolutely. All right. So, if I go here to my windows phone dashboard, okay so you see here I don't sell mine. When I go through it is validation. Submission started package verified submission complete. It hasn't been certified because it was not -- shall because there was an error and I can view it in giving being a stows a document that shows me the problem. In this case I used the windows flag for the icon and app and they say no it locks like microsoft app when it's yours and I have to change the graphics on it so it passes certification. It would say use of brand item global trait mark has -- trademark has been approved by the brand owner. That needs to happen and once that happens you get to go. But it says prohibited application check see Rue -- review the above policy becauseI use something that resembles the microsoft property like logo and they don't allow for that.
>> Got you and that's ought automated or is there a person looking at that.
>> it's both. It's automate and there's actually humans that look at some of this stuff to make sure that we comply with characteristics that are not machine.
>> so.
>> that are executable.
>> ideally I can go write an app and potentially have it publish tonight if it checked out or is there a.
>> okay, yeah. Well.
>> or is there a lag.
>> No depends on how many apps we get. We get 200 apps a day submission something like that 250. And it may take between 3 to 5 days sometimes depends you know.
>> okay.
>> to actually go through the process. If it passes everything, then it's going take less time because you know otherwise they have to let you know something didn't pass and so on.
>> got you. Well, great, thanks so much for spending sometime showing us today. We will be making this presentation available to everyone and also Joel will be sending his slides for everybody. Again, thank you again i just want to let you know about a few events we have upcoming. Today, is double-header day. DigitalGov University afternoon at 2 the dos and don'ts for use in social media a free web nare may 8th at 2 p.m., eastern, store creating engaging content for your site. New media talk and how government uses store fie to reach new audiences and again I will be back here on may 9th at 2 p.m. with the next installment in the mobile development platform web series in apple ios that's a free webinar. Find more info and links how to.gov/training and with that, I will be talking to you all again at 2 p.m. on may 9th. Thank you again Joel and we will talk to you.
>> Sure.
>> bye.
>> take care.
>> the organizer has ended the session and this call will be disconnected good-bye.
[Convenient concluded] -- event -- [Event concluded]