

05192020 Build M365 Rajesh Jha

**Build 2020**  
**Microsoft 365**  
**Rajesh Jha**  
**Monday, May 19, 2020**

[Music.]

(Video segment.)

**RAJESH JHA (EVP, Experiences & Devices):** Hello, everyone. Thank you for being a part of Build 2020. It is such a pleasure to connect with you all virtually today. The fact that this conference can happen right now when we are all working and living apart is really remarkable.

We are truly living in an era of remote everything, and it's clear that this is an inflection point for productivity. Microsoft 365 is the world's productivity cloud, and we aspire to make it the best platform for developers to build productivity solutions for work, life and learning. But we can't do it alone, we need you, and today, you're going to see how we can partner together to create the future of productivity.

Our session today is for every developer, from ISVs building their own apps, to corporate developers creating custom solutions, to citizen developers using Teams and Power Platform.

As an integrated security and productivity solution, Microsoft 365 is a one-stop shop for everything that people need right now, from remote work to telemedicine to learning. But it's not just a set of finished apps and services. It's also a rich developer platform.

This stack diagram summarizes the key components. Our foundation is built on the Microsoft Graph. The Microsoft Graph is the container for all rich data and signals that describe how people and teams work together in an organization to get things done.

Then there's the Fluid Framework, a web-based framework for creating collaborative apps using a new component-based architecture.

And, of course, developers are going to build native apps for Windows, for Mac, for mobile, or use web apps to run across all the platforms.

We'll also talk about Microsoft Teams, both a mission-critical tool for remote work and as a platform for developers to simplify and automate business processes, and have it run on any device.

And finally, we'll share some new innovations to help you build beautiful and productive native apps for Windows.

During today's session we'll step through the components of this diagram and the opportunities for developers. You'll notice that the layers in the model extend across platforms and devices, and that's because Microsoft 365 is designed for people-centric experiences that follow you no matter what device or operating system that you're using.

We are big believers in the power of Teams and we understand that successful teamwork today requires so much more than just connecting during meetings or over chat. We built Microsoft Teams as a platform where people can meet, chat, call, collaborate, all in one place, so that the teams always have the critical contacts that they need to work together, even when they need to work apart.

We've seen unprecedented growth in Teams over the last two months. There are now more than 75 million daily active users, 200 million daily meeting participants, generating more than 4.1 billion daily meeting minutes. As students and teachers move to remote learning, we now have over 180,000 institutions in 175 countries using Teams for education.

But Teams isn't just an app. It also offers rich developer capabilities. Professional developers can build apps for Teams and share them through the Teams app store, and, in fact, over the last two months, the usage of ISV apps in Teams has more than tripled, and over 285,000 new organizations have deployed third-party apps like Poly, Trello, Jura and others, to their users.

In addition, organizations are rapidly building and deploying their own solutions to Teams, including low-code apps built using Power Platform. Power Apps and Teams usage has more than tripled since February alone, with companies like Unilever using Power Apps to stay productive even while working apart.

Put simply, Teams is changing the way people work and learn, and based on what we are seeing from countries that have started to open their economies back up, it's clear that Teams usage will continue to grow, even as people head back to their workplaces.

In a moment, I will turn it over to Archana to talk about the developer opportunity for Teams. But first, let's hear from some of our customers who are using Teams today to keep work moving during this challenging time.

(Video segment.)

**ARCHANA SASEETHARAN (Group Product Manager, Microsoft Teams):** Thank you, Rajesh, and thank you, everybody, for joining us today.

It has been incredibly fulfilling to see how healthcare providers and schools and universities have made Microsoft Teams their workplace. Thanks to developers like you, millions of Teams users are customizing their Teams experience with applications and workflows. Whether you are an ISV or a systems integrator or an enterprise developer, you can extend Teams in several ways to build workflows for your customers or to streamline your own workflows.

You can use connectors to send proactive messages to your Teams channel on code deployments. You can use messaging extensions to collaborate on pull requests without leaving Teams. You can use bots to stay on top of live sites, collaborate and mitigate them. You can use stock apps and personal apps to track and manage all your projects in rich full canvas experiences.

You can use all these extensibility points and more to build similar workflows for several other industries, for healthcare, for financial services, for education, for retail, and many, many more.

You can build all these experiences with the tools and frameworks that you would like. You can use the open dev stack, you can use Microsoft frameworks like the Bot Framework, adaptive cards, Graph APIs, SharePoint framework and Power Platform.

Most importantly, whichever framework you choose to build with, you just build it once, and Teams makes it available across all platforms and across all devices.

Now I'm going to show you the new capabilities that make it even easier for you to build on top of Teams, capabilities that make sure your users can easily discover your apps and help you drive deeper user engagement. Let's take a look.

First, let me show you how you can easily build and publish a Teams app. With Teams extensions for Visual Studio Code and Visual Studio IDE, you can take Teams application development tools to directly where you already work. You can pick the type of Teams app you want to build and then build it. You can create a manifest and package your file, validate it, and with a single tap, you can test your application in Teams across desktop and mobile.

Once you're ready to publish your app, you can either publish your app to the global Teams app store, making it available for all Teams users, or if your app is just for one organization, you can submit it to your IT admin directly for their approval.

Ta-da, that's how simple and fast it is to build a Teams app, all under five minutes.

If you love building low-code/no-code applications, then starting this summer, you can use Power Apps Studio from within Teams. You can bootstrap your app creation with one of several solution templates, or you can create one from scratch. You can easily add Power Apps to any of your Teams channels with one click. You can edit and author these apps inline. You can also leverage the common UI component library to build the Power Apps that have native Teams experiences.

With the Teams extensions for Visual Studio and deeper integrations with Power Apps Studio, you can build and publish Teams apps very, very easily.

But building your app is just the beginning. In order to make sure that your app is easily discoverable by users, we have made it even simpler and faster for IT admins to discover, validate and approve your apps. When you submit a line of business application, your IT admin

will get notified in the Teams admin center. They can then go ahead and review the app details and publish it.

Once an app is published, both line of business applications and ISV applications, IT admins can then easily install and bring these apps for their users, making sure your app is always discoverable.

These new admin center enhancements will be GA'd this summer.

Now that your app is easily discoverable by users, with two new platform integrations you can drive deeper user engagement.

With Azure AAD single sign-on integration, you can enable one-tap sign-in to your app. And with activity feed API integration, you now have a new and simpler way to notify your users across their devices.

Users can also easily customize their Teams workspace with apps with simple pin drag and drop, and very soon, they can also pop apps into their own windows, letting them multitask with your apps any time.

We are very excited that many of you are already integrating with single sign-on and activity app in preview. These integrations will be GA'd this summer as well.

The new developer tools make it even easier for you to build on Teams. The friction-free app submission, validation, approval make sure your app is always discoverable by users, and the new platform integrations let you drive deeper user engagement.

Again, thanks to developers like you, the usage of ISV apps and line of business apps in Teams has been growing rapidly, and several of you are also monetizing your solutions via Teams. Since Build last year, the number of users engaging with ISV and line of business apps in Teams has grown by 750%.

One of our marquis customers, EY, standardized their business processes and workflows on Teams by building six line of business applications. EY employees use these applications every single day to communicate and collaborate, with over 250,000 monthly active users for each of these six applications. Broad Teams adoption has been a critical milestone in the EY digital workplace transformation story.

Now, let's hear from a few partners like you that have built on Teams and delivered customer impact via Teams.

(Video segment.)

**RAJESH JHA:** Well, thank you, Archana. That was great.

I hope you take away just how focused we are on helping you simplify and accelerate Teams app development.

In many ways, Teams is the model for the future of productivity. It brings together everything a user needs to get things done, meet, chat, call, collaborate, create, all in context and all available from any device.

The experiences we create are powered by something that is unique to Microsoft's cloud, the Microsoft Graph. The Graph helps us understand the way people and organizations work on projects, in meetings or as an organization. We can then use these insights to build intelligent, people-centred experiences that work across devices, platforms and applications.

The Graph is the API to your organization. Whether you're building apps for an enterprise or a startup, fluency in Microsoft Graph is an essential skill that prepares you to build the kind of apps and solutions that we believe will shape the future of productivity.

To tell you more about the Graph, let me turn it over to Yina.

**YINA ARENAS (Principal Group Program Manager, Microsoft Teams):** Thank you, Rajesh.

For many of you, Microsoft Graph might still feel like an abstract concept. Let me show you how we use Microsoft Graph across Microsoft 365 to power so many of our experiences on desktop, some old devices, and in browsers, and then talk about it concretely for developers.

I want you to think about Word, Excel and PowerPoint and all of the files that you have stored in OneDrive and SharePoint. Well, those are connected to the Graph. In Microsoft teams, we use Microsoft Graph extensively to create and manage teams, channels, tabs, conversations and more. And so can you, using the API.

And of course, Outlook, which is a rich source of productivity data. Microsoft Graph has all the emails, meetings, tasks, and you can access that with the API, and even some of the intelligence behind it like, for example, the focused inbox.

Microsoft Graph has so many relevant schemas of data that allows us to build people-centric experiences and that you can use to build rich, contextual, relevant experiences that are uniquely focused around users and organizations.

As a developer, you can call into Microsoft Graph using simple HTTP requests and receive standard JSON formatted responses. For those of you who prefer to use the SDKs, well, we have them available in a variety of languages.

And we have lots of developer tools available. Our API playground, the Graph Explorer, is getting new functionality, like this new way to see all of the permissions that are required for each API, or see the code snippets using the SDKs.

And one of our favorite initiatives, the Microsoft Graph Toolkit, has a new developer playground and, of course, super easy to use web components that are framework agnostic, and that you can add to your applications with just a few lines of code.

For developers, I'm going to summarize the opportunity in two things. First, you can use Microsoft Graph to extend Microsoft 365 experiences. For example, you can build an app for Microsoft Teams that is tailored to a particular customer need. And two, you can create your own applications powered by the data and intelligence behind Microsoft Graph. It can be a web application, mobile application or many others.

To learn more about Microsoft Graph and getting started with developing with Microsoft Graph, go today to [Graph.Microsoft.com](https://Graph.Microsoft.com).

**RAJESH JHA:** Hey, thanks, Yina.

Getting started at [Graph.Microsoft.com](https://Graph.Microsoft.com) is the first step. Now can you tell us about the work we are doing to help developers drive adoption and to connect them to our enterprise customers?

**YINA ARENAS:** Well, Rajesh, because there's so much power even in a simple dataset, it is really important that people understand that these apps are trustworthy. Customers really want to know that the apps that they install come from an authentic source and that they're built according to industry standards and best practices.

Today, we're announcing the addition of publisher verification and ramping up our app certification process for Microsoft 365 apps. These apps have now badges that customers can see and indicate that Microsoft has verified or certified the application. And then for IT admins, they can configure policies to give these apps preferred treatment in the organization.

**RAJESH JHA:** Yina, we know that the Graph contains a huge array of schemas that developers should consider essential. And it's really important that we are leaning in with the developer community to emphasize the trustworthiness of our ecosystem, so that was great to hear.

I'd also like to bring up the point that the Graph is more than schemas, it's about services, and we have a set of capabilities that can help developers build productivity solutions that scale to meet the challenges that they are facing as remote work and efficient workflows become not just important, but critically essential to our customers.

**YINA ARENAS:** Yep. We call these capabilities Microsoft Graph Services. These are productivity and security services powered by Microsoft 365 and Azure that can help you build high value applications.

I'll share three kinds of services with you today. First, we have connectors. With connectors, you can move data in and out of Microsoft Graph. For example, if you want to augment one of our Microsoft 365 services like Microsoft Search, you can use connectors to index data that is in a line of business service or in another cloud share or even in your file sharing system on-prem. And then you can participate in all of the Microsoft 365 search experiences. Then if you want to

take data from Microsoft 365 and move it out into Azure to do some post-processing there and build your own AI experiences, you can do that with connectors as well.

Second is a set of services that Microsoft Graph has that help you manage compliance and protect your organization from data leakage and loss. Recently, we started offering developers this set of new security and compliance features, like the introduction of a new Teams chat web API that expands our capabilities in this area.

And third, we have the knowledge generation services. This is an area where we really see an opportunity for you as a developer to use Microsoft Graph to its full potential.

One of the best ways for me to show you how powerful this is, is to give you a very quick tour of Project Cortex. Project Cortex uses AI through Microsoft Graph data in your organization to automate content capture and categorization. Cortex then organizes that content and expertise across your teams and systems of record, interrelated topics such as projects, and then creates a knowledge network just for you and your organization.

OK, so let's see what Project Cortex makes possible. Let's say that I am a recently hired employee who has just received an email about a new project. In the message text you can see we have this word highlighted, core. I don't have previous context about core, but Project Cortex has automatically identified this as a topic in the knowledge network. And when I hover over the word, this topic card appears. From there, I can quickly get an overview of the project, as well as links to relevant people and documents.

All of this was automatically generated by mining the knowledge network behind Microsoft Graph. In other words, Cortex generates knowledge tailored specifically and relevantly for you and your organization.

Now, the exciting part for you as a developer is that this summer, we're adding the new taxonomy and knowledge API behind Project Cortex to the Microsoft Graph, so that you can customize the way knowledge in your organization is surfaced and shared.

The next thing that I want to share with you is a customer story. It is about how the San Francisco Conservatory of Music is using Vegas Stream, software built by our partner Magix. This software is connected to Microsoft Graph in amazing ways. Let's take a look.

(Video segment.)

**YINA ARENAS:** It's a great story and one that really resonates with these times, and I hope that it inspires you to build Microsoft Graph powered experiences in your own applications.

And I have one last experience to share with you. If you watched my friend Scott Hanselman and his developer keynote, you noticed that he had an IoT device, a LightFX light like just like the one I have behind me.

This light is actually powered by an app using Microsoft Graph. The app gets my presence status information from Microsoft teams, whether I'm using Teams on the web, on my laptop or on my phone. It's using a simple call to the Microsoft Graph, and this light is helping me manage my interactions with my family as I balance working from home every day.

It's a simple but powerful demonstration of the Graph capabilities, an IoT device that now knows two things: who am I and what am I doing.

I'm Yina and my kids are ready for this to be done. Back to you, Rajesh.

**RAJESH JHA:** That's awesome. Thanks, Yina.

At Microsoft, we're using the Graph to create valuable, differentiated experiences across our products and services, and you can start using it today to enrich your solutions and create more personalized and productive solutions.

Teams is the collaboration tool for remote work and learning and the Graph connects your apps to valuable business context.

Well, what about content-centric collaboration, the heart of so many business processes? That's where Fluid Framework comes in. Fluid Framework is built both for end users and for developers. For end users, Fluid is a new way to collaborate with others. It includes new Fluid components like tables and lists and agendas that can be inserted directly into emails and chats.

These Fluid components work a lot like a shared document. Multiple people can edit them simultaneously and all the changes are broadcast to all the editors. But unlike a document, a Fluid component is fully rendered inline. You don't click on a component and go to another browser tab. You see the entire component in context and can immediately start interacting.

For developers, Fluid is a web-based framework that you can use to instantly make your apps collaborative. It includes distributed data structures that perform low latency synchronization and has a relay service to connect different endpoints.

If you simply replace your static data structures with Fluid data structures, your app instantly supports real-time collaboration. And using Fluid components, you can embed custom app experiences in other apps, both inside of Microsoft 365 and beyond.

I'd like to introduce Sam, a developer on Fluid Core team, to show you how you can use Fluid Framework experiences. Sam?

**SAM BRONER (Software Engineer, Fluid Framework):** Thanks, Rajesh.

I want to show you all two things today: first, how easy it is to build natively multiuser apps with Fluid Framework, and second, how you can embed and share your live web experiences everywhere with the help of Fluid.

We architected Fluid for super low latency distributed systems, and yet I have never made more impressive web apps more quickly. We created a set of distributed data structures with the merge logic implemented in the client JS that handle the difficulty of distributed data merge for you.

Today, we're going to upgrade a simple web app with Fluid distributed data structures and Fluid components. There's no server code needed and we're going to do it all in the next two minutes. Let me show you.

Now, for a hackathon we recently ran, I wanted to build a simple hack tracker to manage all of our incoming projects. The hack tracker has a list of our products to the left and a message board to take notes in the middle.

I built this very simple application with React, but of course, you could use any frontend framework.

While our app lets me take notes locally, it's not wired to anything yet. I also can't share my notes or interact with coworkers during the hack week. It's easy to add all of this functionality with Fluid Framework. Fluid distributes server-side data and logic to the browser and gives me access to distributed app state locally. Again, there's no server need. These two attributes, combined with our component model, will make upgrading our hack tracker much, much easier. It's just waiting to be wired up.

By replacing this JavaScript map with a Fluid shared map, I can quickly bring the application to life. I also have to listen to events on our shared map so that I can update our product list with changes from other users. Now we can reorder and add new projects collaboratively.

Let's look at our message board. A tremendous amount of work has gone into making excellent open source rich text editors. Let's take Draft.js, a great text editor, and make it collaborative. This typically requires a purpose-built service integration, but we can just use Fluid Framework as the collaborative data layer.

I'll have to write two code snippets here, about 100 lines of code total, one to take changes from other users and replicate it into Draft.js's editor context, and another to take our editor's context and replicate it into Fluid. This is similar to the project list. We need to change the distributed data structure state locally and then listen for changes from other users. I also included some model adapter code.

Now we have live notes during our hackathon week, built using Fluid and excellent open source tooling.

While we'll stop here. I could take the time to polish this adaptor up and publish it to MPN. Because the server code lives in the client with Fluid, any developer could install our Fluid Framework Draft.js component and deploy it, batteries included.

We'll support embedding of Fluid components across M365. For example, I can paste my hack tracker into Teams and share it, always up to date, with everyone in the hackathon.

Fluid Framework SDK will come with all the tools you need to make your own collaborative applications, but it also has a component model that makes it easy to reuse and embed your live Fluid parts within other Fluid enabled Web apps.

As you can see, we had a group of amazing projects during the hackathon week. People used Fluid to quickly bootstrap a group of live tools such as the Python environment for the classroom, a herd management tracker, and a collaborative MIDI keyboard. The hack tracker code will be available as part of the Fluid Framework SDK.

One of the biggest users of the Fluid Framework is Office. Now we'll hear from Maya and Dan, who will show you how we're planning to integrate Fluid within Office and across applications.

**MAYA RODRIG (Principal Program Manager, Fluid Framework):** Thanks, Sam. Let me show you how Dan and I used Fluid Framework across the Microsoft 365 ecosystem.

I'm writing an email to follow up on next steps. I can pop open this new menu of collaborative Fluid components that's available to me in Outlook and Teams. These components are all built on the Fluid Framework that Sam just showed us, so his component could appear here, too. Many of these were designed with specific user intents in mind, like action items.

**DAN COSTENARO (Principal Program Manager, Fluid Framework):** In that email I just got from Maya, I see she's editing and it's easy for me to edit, too, right here in Outlook.

**MAYA RODRIG:** Now I want to share the action items with another group of people. In this case, we use Teams for quick collaboration, so I can paste the action items right here in our chat. And because we're reusing a Fluid component, we have a single source of truth, no matter which app or device people collaborate on. And it looks like everyone's jumping in to contribute.

**DAN COSTENARO:** Those Fluid components are stored in SharePoint files. Fluid Framework's distributed data structures and component model provide high scale, low latency, single source of truth within Teams, Outlook and Office, to power everyday scenarios. We're excited to co-create the future of collaboration together with you.

**RAJESH JHA:** Thanks, Sam, Maya and Dan, for showing Fluid Framework in action.

As you can see, Fluid Framework opens a whole new world of possibilities. As a developer, you can make any app collaborative by simply adding Fluid data structures. And as an end user, you can start using Fluid components and emails and chats to collaborate in context.

But the power of Fluid to advance the state of art and collaboration will really come from the richness of the framework. And to that end, we are excited to announce that we will open source the Fluid Framework.

We can't think of a better way to invent the future of collaboration than, well, to collaborate with you all and invite you to contribute your feedback, your features and your pull requests. We're excited to see what we'll be able to create together.

And we've been working with partners, large and small, to get their feedback, and one partner, Templafy, focuses on making document management easier. Let's see what they built with the Fluid Framework.

(Video segment.)

**RAJESH JHA:** Now, let's talk about the operating system that so much of this development and usage is taking place on, Windows. Earlier this year, we announced that there are now more than a billion monthly active Windows 10 devices, across 200 countries from around the world.

And we are seeing Windows play a critical role in these changing times, helping people work, learn and stay connected from home. In fact, people are spending more than 4 trillion minutes on Windows 10 each month, and that is 75% more time than last year.

And of course, people are engaging with Windows 10 through apps and tools built by our incredible community of developers. But we also hear the developer community tell us that it's not always easy to reach all billion of the Windows 10 devices at once.

So today, we are announcing some plans to make it easier for developers to create solutions that work for today and tomorrow.

I'd like to welcome Kevin to provide details about how you can build rich apps to reach these billion devices.

**KEVIN GALLO (CVP, Developer Platform):** Let's take a deeper look at Windows and Microsoft 365 and how you can build more engaging, connected, and, of course, beautiful experiences.

Windows 10 presents an opportunity to reach more than a billion devices, from lightweight, always connected ARM devices, to the most productive workstations on Earth.

We all want to build apps that people love and that we're proud to ship because they look and work great, but we've heard your feedback that this can be a heavy investment. You want a common platform and a less complicated way to design apps for Windows.

I want to share three improvements we're making to the Windows platform. First, we're unifying app development across the billion Windows 10 devices for all your current and future apps.

Second, we're creating new opportunities for you to build connected apps using Microsoft 365 integration in the Windows experience.

Third, we're leaning into the cloud and enabling new scenarios for your Windows apps.

Let's start with how we're making it simpler to build a single Windows app that reaches all Windows 10 devices.

Now, over the last couple of years, we've been breaking down the barriers between Win32 and UWP. We're expanding this effort even further now and finally giving it a name, Project Reunion.

Today, apps generally call directly into APIs in Windows, whether they're desktop Win32 APIs or UWP APIs. With Project Reunion we're making many of those same Win32 and UWP APIs available in a better way, by unifying the existing APIs and decoupling them from the OS. This allows you to mix and match from the APIs that work best for your app and adopt new capabilities on demand. You can also unlock the latest APIs without waiting for your users to update their OS.

We do the heavy lifting by pollyfilling gaps down level as needed, and we've already begun with APIs like WinUI 3, WebView2 and MSIX. In fact, I'm proud to announce we have new previews of WinUI 3 and WebView2 available today.

We're also excited to see that companies like Esri, a leader in geographic software, are already actively building on these previews, and we want to invite you to join us in this transformation.

To really now understand how Project Reunion works, let's see it in action. Here I'm building a sample app for farm management. This could be any Windows project, UWP, WinForms, WPF, React Native or C++. But I chose to use an existing desktop app.

I also wanted to just update the user experience and not the underlying business logic. So all I need to do is add the NuGet packages and the APIs will pull from those instead of from the platform SDK.

In this specific instance here, I'll expand my packages. You'll see I have WinUI, I have WebView2, and I have some packages here that allow me to use .NET with these APIs.

Now, these packages include the same APIs as the platform SDK, so your code doesn't have to change. You also get to release the app to all versions of Windows 10 like you did before, since it is decoupled from the OS.

Now let's move on and take a look at some of the changes I made to the experience in the app itself. You'll see here when I launch it, it gives me this great animation. It'll automatically log me in, just like I expect. And then I'll go and I'll see a list of the farms that I've been working on. Click one and you see all the details all laid out in a more productive and easier to use experience. I chose to build this with WinUI because it's one of the first Project Reunion components we're now making available decoupled from the OS.

But before I go any further into the app, I want to show you the second improvement we've made, so your apps can use Microsoft 365 integrations like Microsoft Search and the Microsoft Graph.

Now, you'll see here I had a nice link on my desktop, easy to launch, but that is not realistic in the real world. We all use too many apps and struggle to find things. Microsoft Graph and Microsoft Search can draw unique connections between your people, files and tools. And because search is integrated in Windows, you can find these through a familiar interface.

For example, Carol showed me some information the other day in the sample farm app, but I wouldn't always remember which app or dashboard. Normally, I'd have to ask her, but now I can just search for Carol, and the Graph knows that this is the Carol I work with, so she shows up. I can see her organization, her office location, and even the files that we've been working on together. There it is, the file that we were working on that she told me about. I can click on it, and it relaunches me right back into the application at the dashboard we were looking at. All of this is because of the power of the Microsoft 365 platform. As a developer, I didn't have to do any work to actually get this functionality.

OK, let me continue now with the updates I made to the app itself. I wanted this user experience to be productive with touch, mouse, keyboard and pen. For example, I can go swipe, and I'll be able to go and let me swipe this other one, and there you go. I can go change it. I can also go and edit the fields and use my mouse and my keyboard to be more productive when entering data.

Here I can just tab around. It tabs nicely. I also can get inline input validation. For example, I'll just change the location to something that's invalid, try to hit save. It will tell me the address is wrong.

Now, a nice little feature that we've added to our number control is the ability to enter math equations. Here, if I want to go figure this out, I wouldn't have to do the math myself, and right here in line, I get the calculation, great experience for your customers.

Now let's go take a look at the code in Visual Studio here and show you just how easy it was to do that. As you can see, I'm writing the UI layer in .NET, but more specifically, the latest version, .NET 5. However, my business logic was written as a C++ ETL com object that I could have written 15 years ago. And I'm able to reuse it here, without any modification, allowing me to focus on the user experience. That compatibility is one of the promises of the Project Reunion model.

Now, to get that great swipe capability, here let me go to the stage control, and yep, I've added the swipe control and I have two items that show up on each one of these stages. And all I had to do is I have done and start, and have two different event handlers here and here, and I'm done, everything else built right into the controls.

Same thing for the number box. If I want to go get all that great mathematical expression, all I have to do is change one property saying accept it and that will work.

These are just some of the controls that we're releasing with WinUI that can help you quickly build a fast and productive experience.

Now, there are also times when you want to use web technology so you can share code across platforms and with the browser. I'm going to head over to the document tab here and show you that this UI on the right-hand side here is all written in HTML, CSS and JavaScript, and it's made possible with the WebView2 control, built on the new Microsoft Edge.

With WebView2 I can even do things like click and show PDF files. I'll just click on one of those cards, instantly shows up with a PDF inline here with a great experience, just like I would expect on a browser.

Let's go look at the implementation for this. I'm back here in Visual Studio. I'm going to go to my card view. And here with one line, I say I want a web view. I give it the URL, I get everything that you saw there, including all that great PDF support.

Previous Windows web controls were locked to a single version of Windows. As part of our Projection Reunion vision, WebView2 is now decoupled from the OS, bringing the power of Edge and Chromium to the full spectrum of Windows apps.

Starting today, we have preview versions of the WebView2 control available for C++, WinForms, WPF and WinUI. We're rapidly approaching our first stable release, and several product teams within Microsoft, including Office, are prototyping with the WebView2 previews.

Now, finally, let's talk about the third improvement and how you can lean into the cloud to enable new scenarios for your Windows apps with Windows Virtual Desktop or WVD.

WVD empowers remote work and education solutions for your Windows apps on iOS, Mac, Android, Linux and of course, Windows. Let me show you how the WinUI experience in my sample app adapts when running on an iPad.

Here you'll see I'm running the exact same app that I was running on my desktop before, except it's now adapted to the 11-inch iPad, so it reflows the content on the right-hand side to look more natural. I can scroll and see that I have all the same widgets. I also get to use all those great swipe controls, just like I would expect before. I also get the ability to use my cameras and microphones and other peripherals, so it feels natural for me on this device.

Now, WVD also allows you to benefit from the amazing scalability of Azure, so you can quickly add users and optimize your workloads.

Because scalability is so important, we're working on a new feature we call MSIX AppAttach. With this, WVD will be able to deliver apps faster and you can better optimize costs. By adopting MSIX for Windows desktop, the same investment will soon bring even more benefits when running your app on Azure with WVD.

So today I've shared with you just a few of the improvements we've made to Windows, directly inspired by your feedback. Please continue to work with us so that together we can build the best platform for your business. Thank you.

**RAJESH JHA:** Thank you, Kevin.

It's amazing to see these innovations in Windows. I look forward to seeing all the great things you and our developer community are going to go build next.

We've covered a lot of ground in a short time. I want to thank my colleagues for joining and sharing what they are working on.

So we've seen how Teams is powering remote work and learning across the world, and how you can build Teams apps to simplify and automate business processes.

We've looked at how Microsoft Graph is a powerful API to your organization and how you can use it to build personalized productivity solutions.

And we've examined how Fluid Framework is opening a whole new world of possibilities and how it will give us an opportunity to work together to create the future.

And finally, we watched it all come together in Windows with new innovations that will help you build beautiful and productive native apps.

It's an exciting lineup of new innovations, and we can't wait to see what you build with it.

So to close, let's hear from a few developers who are doing some pretty amazing things with the Microsoft 365 platform.

(Video segment.)

END