



Edizione 2019

VCL E WINDOWS 10

Nuovi componenti e servizi



Luca MINUTI
DEVELOPER

email

Luca.minuti@gmail.com

GITHUB

[HTTPS://GITHUB.COM/LMINUTI/](https://github.com/Lminuti/)



LUCA MINUTI

PROGETTI

- WIRL
- OPENSSE
- WKHTMLTOX

CORSI

- DELPHI - WEB - REST
- SENCHA EXTJS



Edizione 2019



MATTIA VICARI
DEVELOPER -
INFO.C.E.R.

email

3DCRYPTER@gmail.com

GITHUB

[HTTPS://GITHUB.COM/MATTIVICARI/](https://github.com/mattivicari/)

WEB

[HTTPS://3DCRYPTER.IT/](https://3dcrypter.it/)

 **DelphiDay**
italian conference

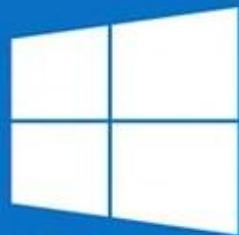
PADOVA
EDITION

Edizione 2019

 **wintech**
italia

AGENDA

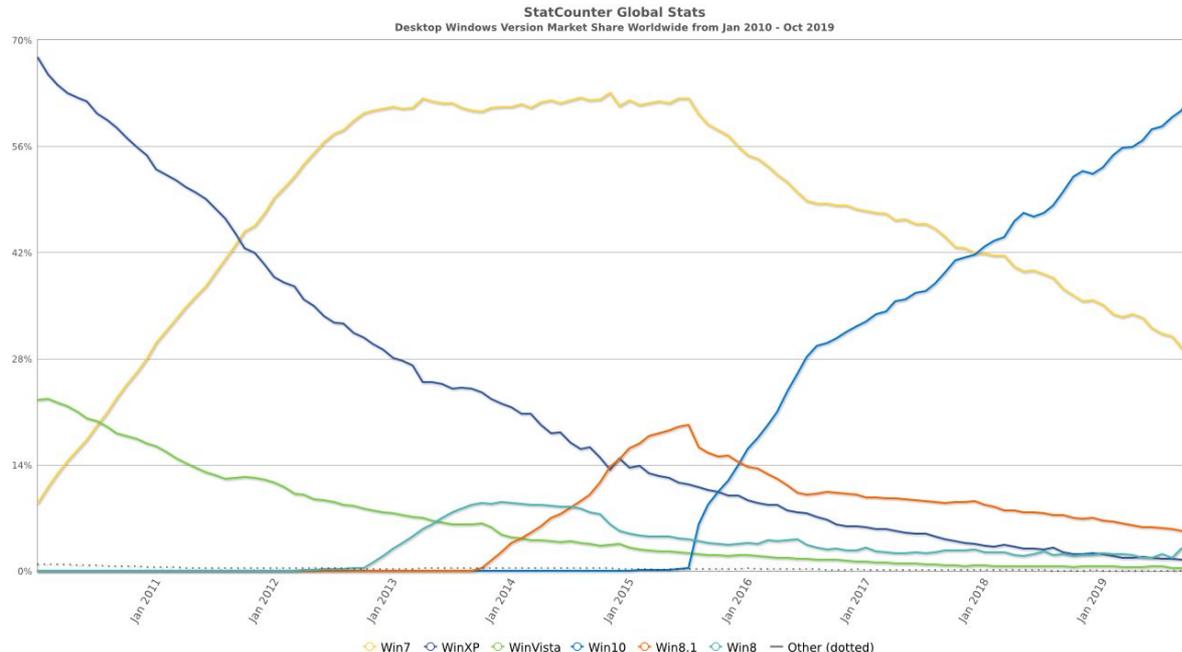
- Introduzione a Windows 10
- Fluent design
- Nuovi componenti VCL e terze parti
- Nuovi layout
- Supporto High DPI
- WinAPI, COM, WinRT
- Microsoft Store



Windows 10

WINDOWS 10

- Nasce il 15 luglio 2015
- Sorpasso di windows 7 nel 2017
- Al momento è più popolare di tutte le altre versioni di windows messe assieme



INFO 1/2

- Riceve nuove build su base continuativa (senza costi aggiuntivi)
- Supporto per le app universali (**UWP**)
 - ◆ PC, tablet e smartphone (ovviamente MS)
- interfaccia ottimizzata per touchscreen
- Nuovo browser Web **Microsoft Edge**
- Gestione dei desktop virtuali

INFO 2/2

- WSL: windows subsystem for linux
- DirectX 12
- .NET Framework 4.6 (alla data di pubblicazione)
- Rimossi i driver per unità **floppy** esterne!





Fluent Design System

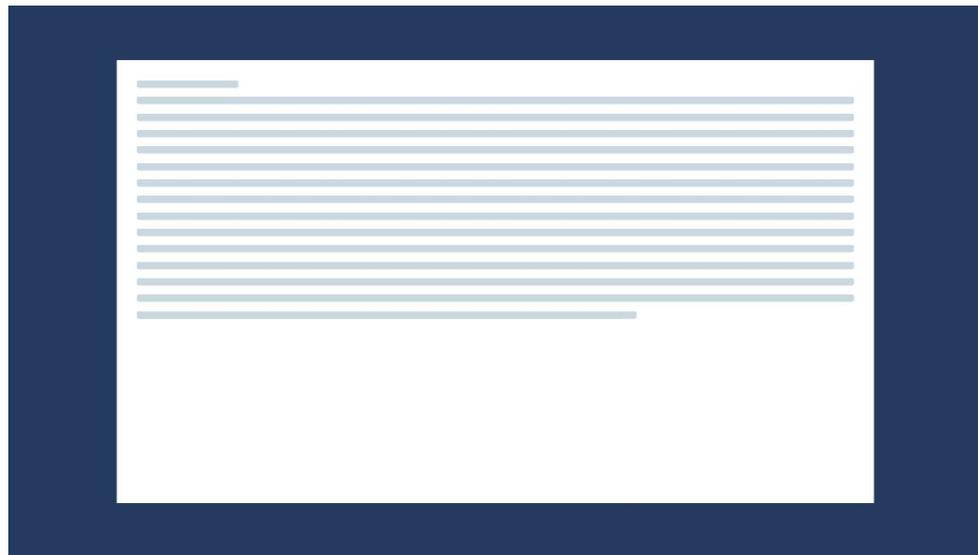
PRINCIPI BASE

- Adattabile a vari dispositivi
- Anticipare le richieste dell'utente
- Intuitivo attraverso luci, forme, movimenti, ombre, profondità



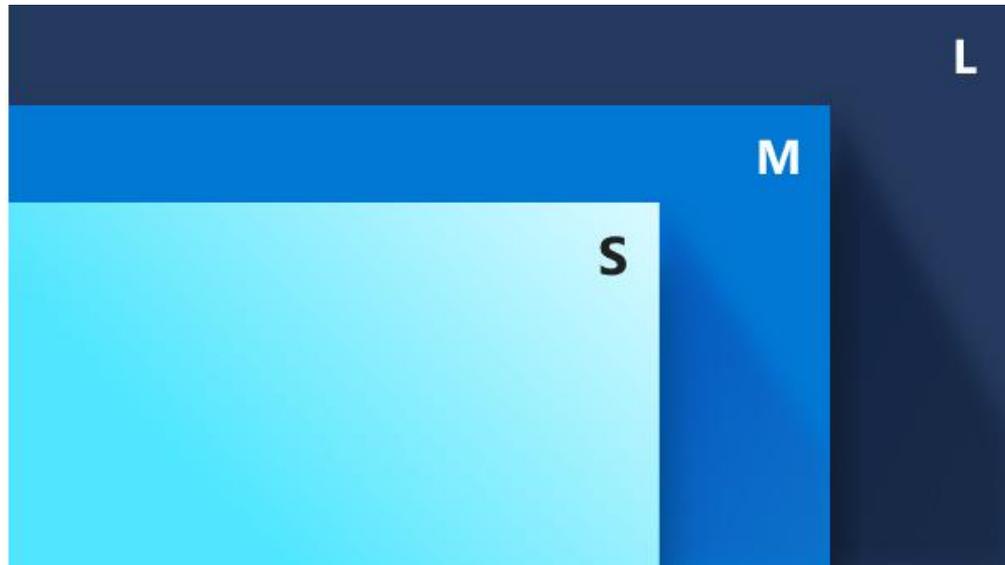
RESPONSIVE

Invece di creare un design fisso adattarsi alle varie risoluzioni di sistema



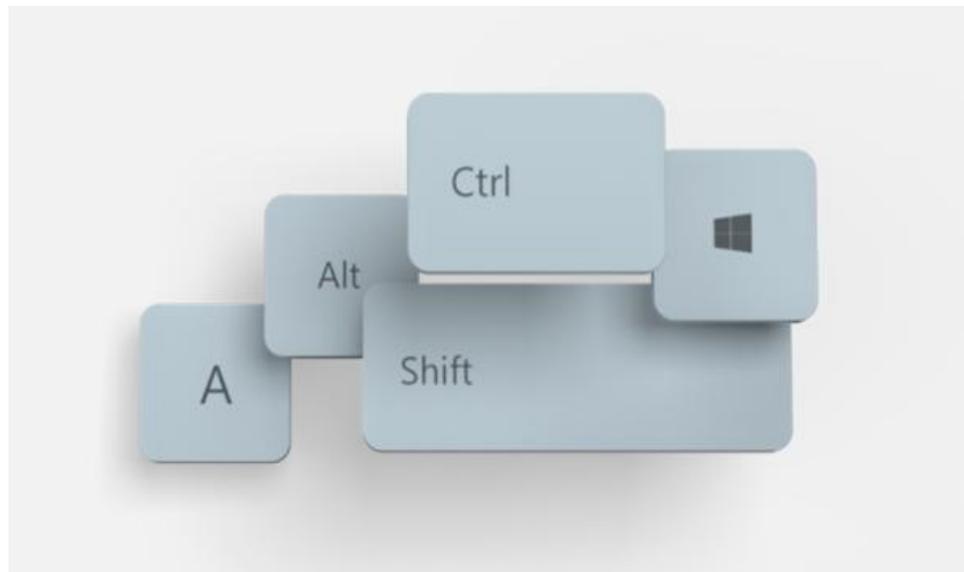
BREAKPOINTS

Cercare le
dimensioni chiave
dove effettuare un
cambiamento più
drastico



INPUT

Adattare l'input al tipo di dispositivo: mouse, tastiera, penna, tocco...



IMMERSIVA

Reveal focus

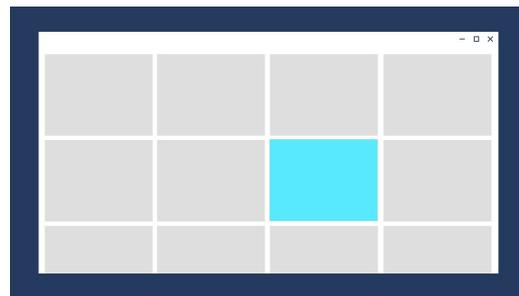
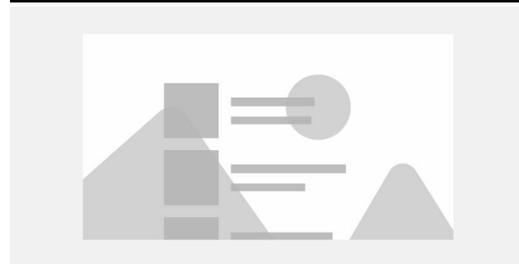
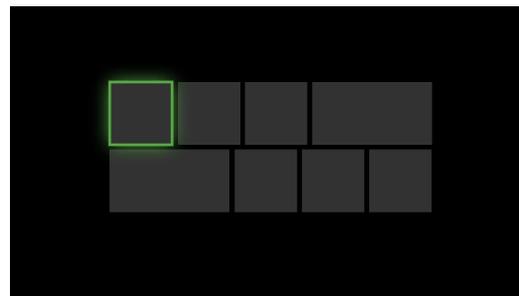
Rendere evidente dov'è il focus dell'applicazione

Effetto parallasse

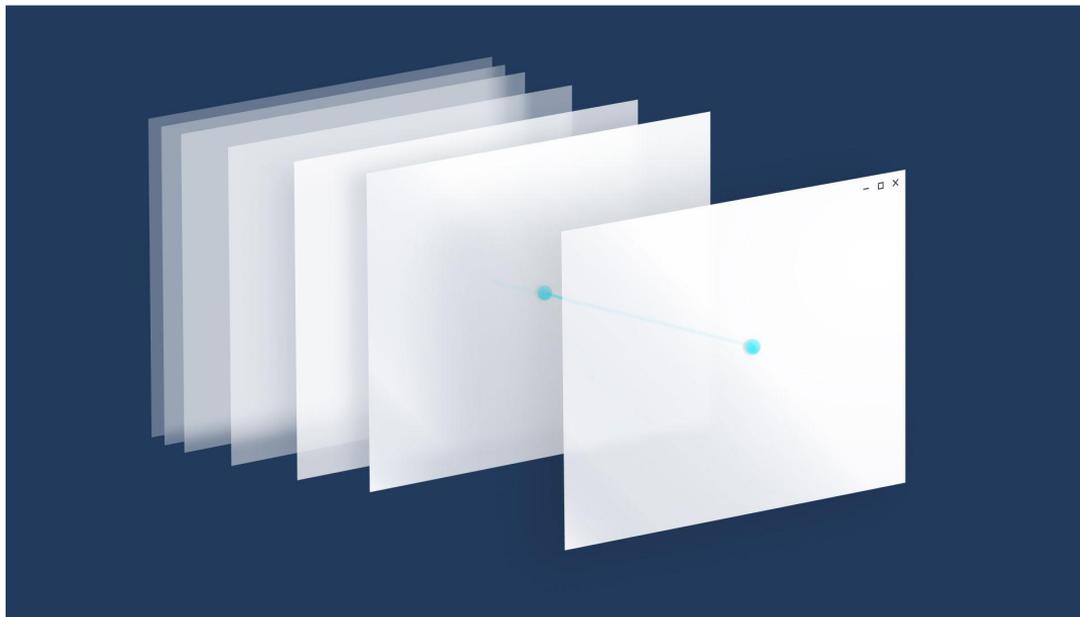
Creare l'illusione di profondità

Effetti di movimento

Per dare un senso di continuità e contesto alle operazioni



NAVIGAZIONE



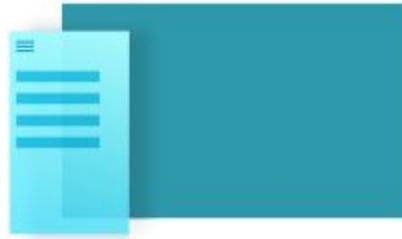
La navigazione si basa sul concetto di pagine

<https://docs.microsoft.com/it-it/windows/uwp/design/basics/navigation-basics>

NAVIGAZIONE



L'utente naviga di pagina in pagina



Un menù consente di saltare ad una pagina specifica



Tutta l'app risiede su un'unica pagina

NAVIGAZIONE

Consistenza

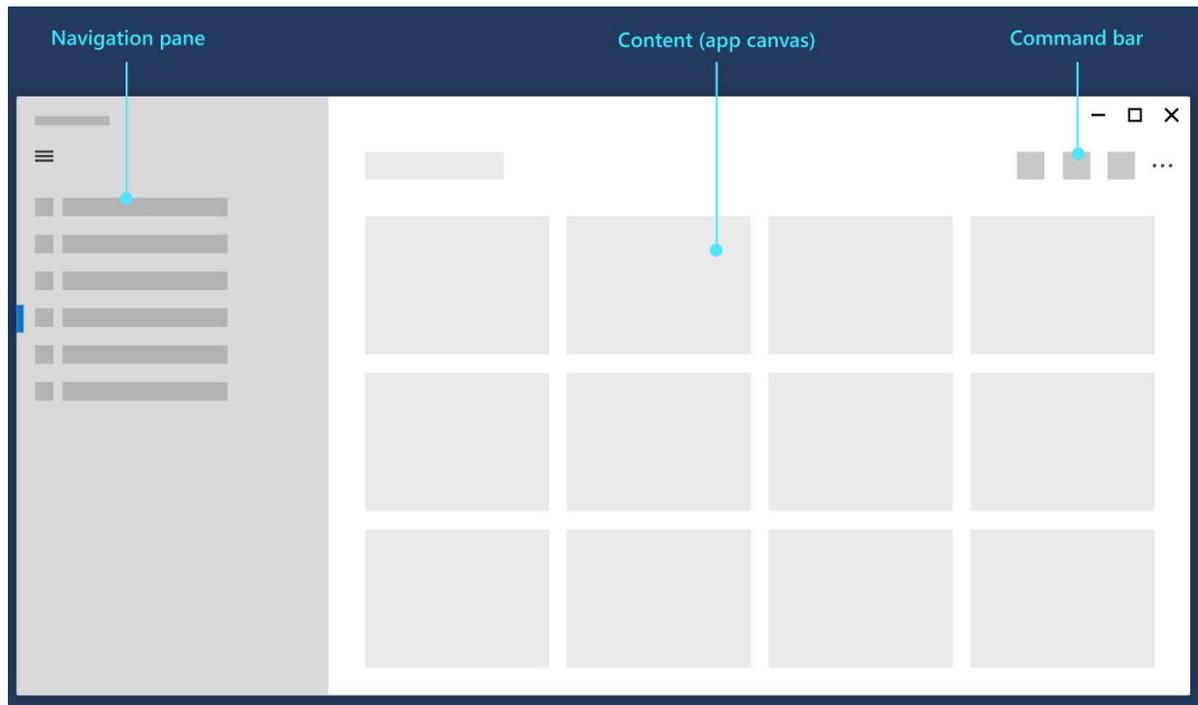
Assecondare le aspettative degli utenti

Semplicità

Non fare più di quello che serve

Comprensibilità

Fornire percorsi e opzioni chiari



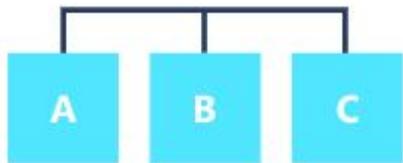
NAVIGAZIONE

Raccomandazioni generali

1. Per ogni pagina pensare perché l'utente è lì e cosa deve fare
2. Evitare gerarchie di pagine troppo profonde
3. Evitare il “pogo-sticking”. Quando un'operazione richiede di salire e scendere più volte nella gerarchia di menù

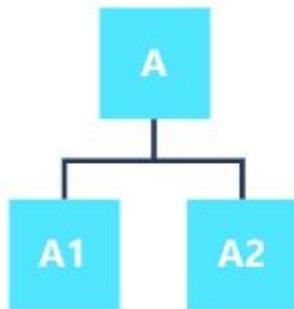


NAVIGAZIONE



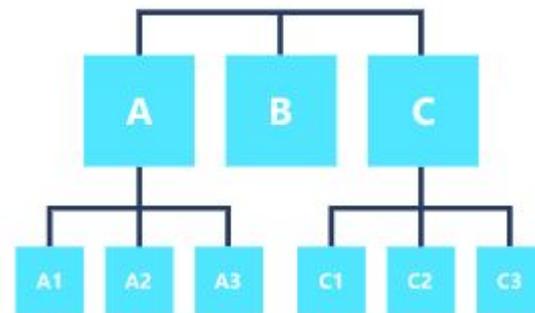
Piatta

Si può navigare tra una pagina e l'altra in qualsiasi ordine.



Gerarchica

Buona per strutture complesse con relazione padre/figlio. Attenzione a non esagerare con i livelli.



Ibrida

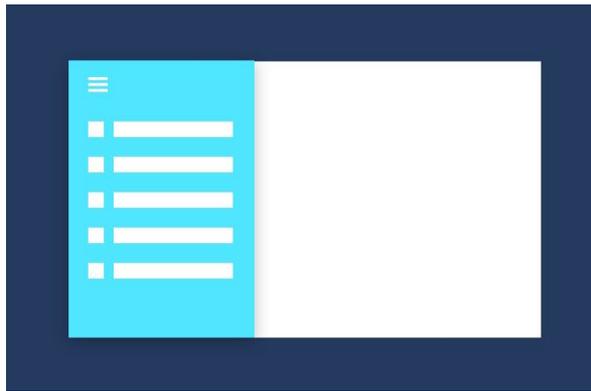
Le due impostazioni precedenti possono anche essere combinate.

NAVIGAZIONE



Tab navigation

- Switch tra i tab frequente
- Più spazio per il contenuto



Le navigation

- Molte opzioni di navigazione
- Lo switch non è molto frequente



Master/detail

- Quando c'è una relazione padre/figlio e lo switch è frequente
- Lista contatti/data entry

TYPOGRAPHY

Segoe UI

- L'applicazione dovrebbe usare un unico font
- Raccomandato Segoe UI
- Installato con Microsoft Office 2007 e Windows Vista

Type	Weight	Size	Line height
Header	Light	46px	56px
Subheader	Light	34px	40px
Title	Semilight	24px	28px
Subtitle	Regular	20px	24px
Base	Semibold	14px	20px
Body	Regular	14px	20px
Caption	Regular	12px	14px

TYPOGRAPHY



Double line list item

A single line of text can clip at t



Triple line list item

Two lines of text can wrap on the first line and clip on the sec



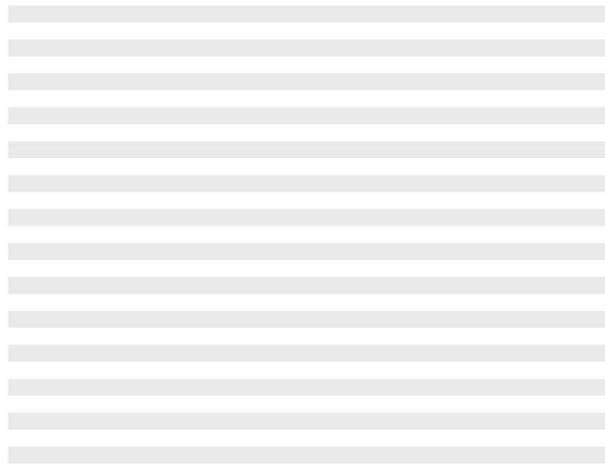
Alan Granger

Hey! I got the tickets.
He doesn't need to worry about



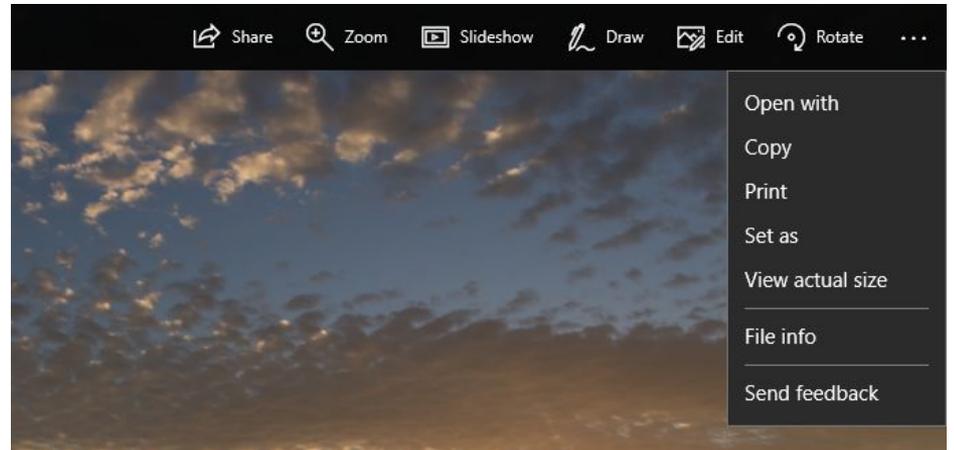
Shiela Tonnes

Has Jon made reservations?
We need to make sure that the



COMMAND BAR

- Sostituisce la vecchia toolbar
- Comprende le funzioni più usate e un “see more” button formato da tre puntini (⋮)



ICONS

Icone predefinite

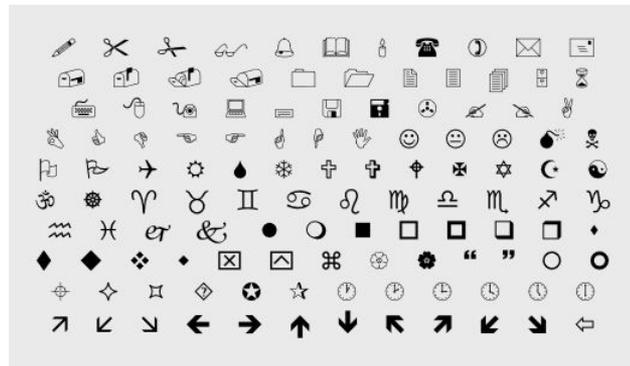
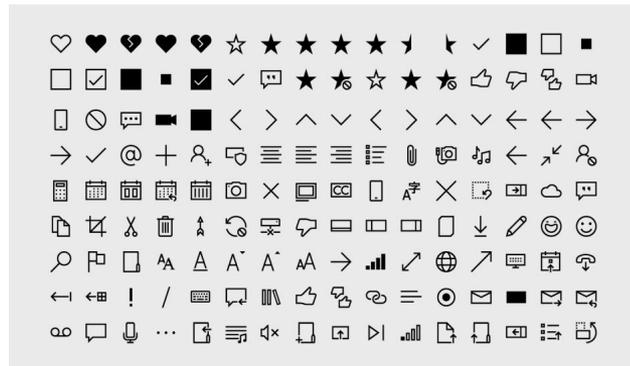
Il set di icone predefinito è quello presente nel font di sistema **Segoe MDL2 Assets**

Altri icon font

E' possibile usare altri font di icone come **Wingdings** o **Webdings**

Immagini vettoriali

Negli altri casi per mantenere la scalabilità delle icone bisognerebbe usare dei formati vettoriali (SVG)



CONTROLS

A System accent color: #0078D4

Buttons

- Enabled button
- Disabled button
- Toggle button

Calendar Date Picker

Label title

mm/dd/yyyy

Hover

mm/dd/yyyy

Disabled

mm/dd/yyyy

February 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

Combo box

Label title

Placeholder text

Hover

Placeholder text

Disabled

Placeholder text

Microsoft

Windows

Office

Microsoft

Windows

Office

Textbox

Label title

Placeholder text

Hover

Placeholder text

Disabled

Placeholder text

Typing

This is text.

Password

.....

Toggle switch

Off Disabled Off

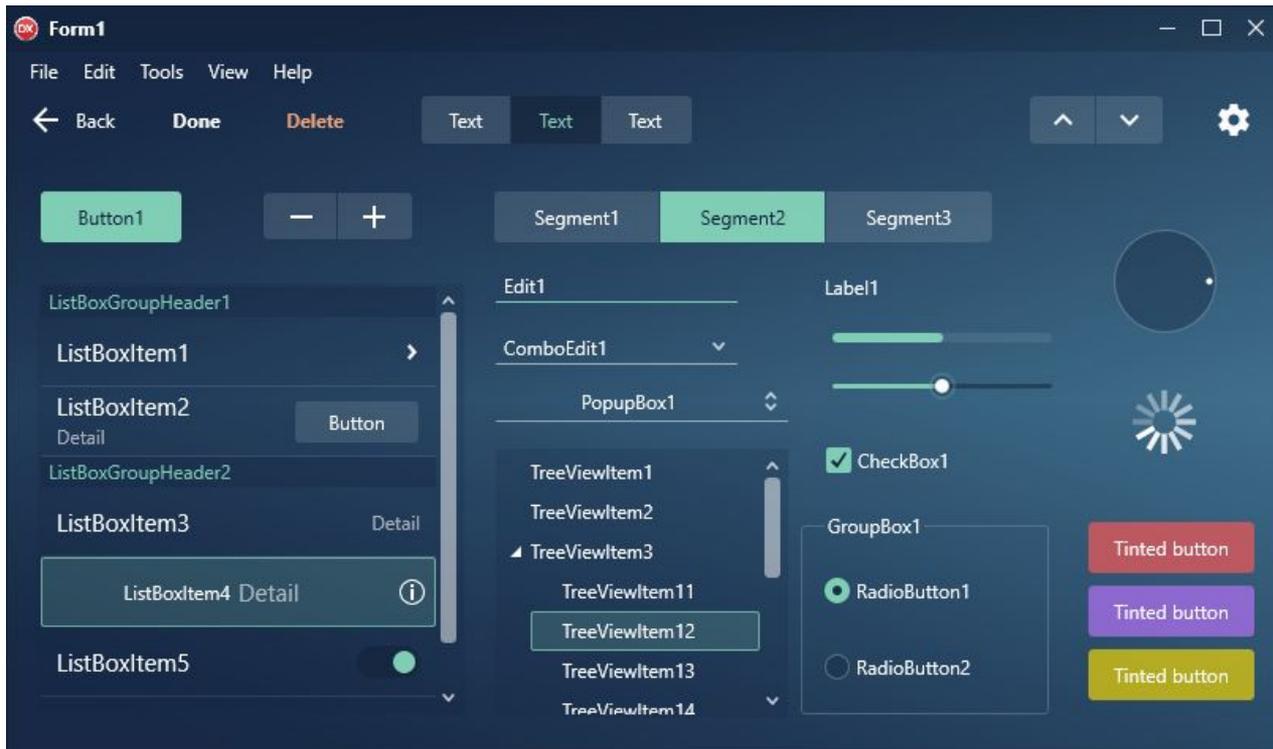
On Disabled On

I controlli sono elementi dell'interfaccia utente che permettono di eseguire delle azioni



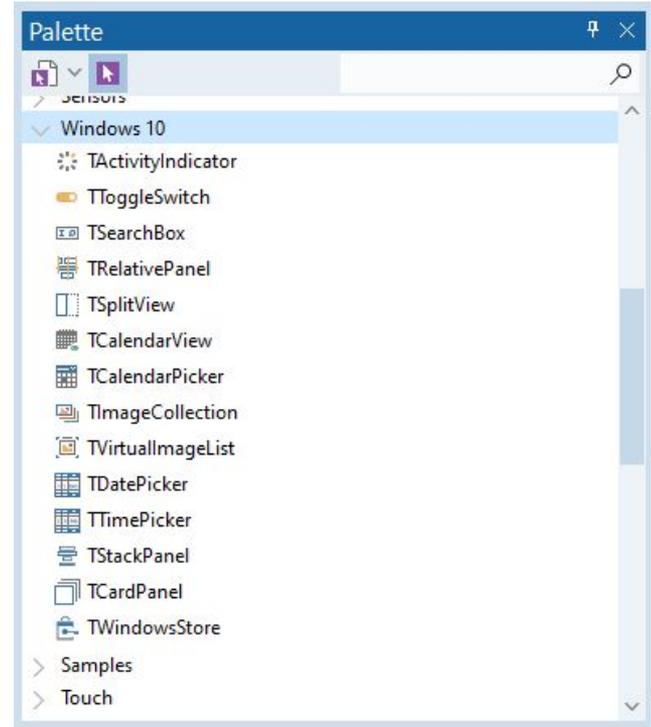
E IN DELPHI?

COMPONENTS



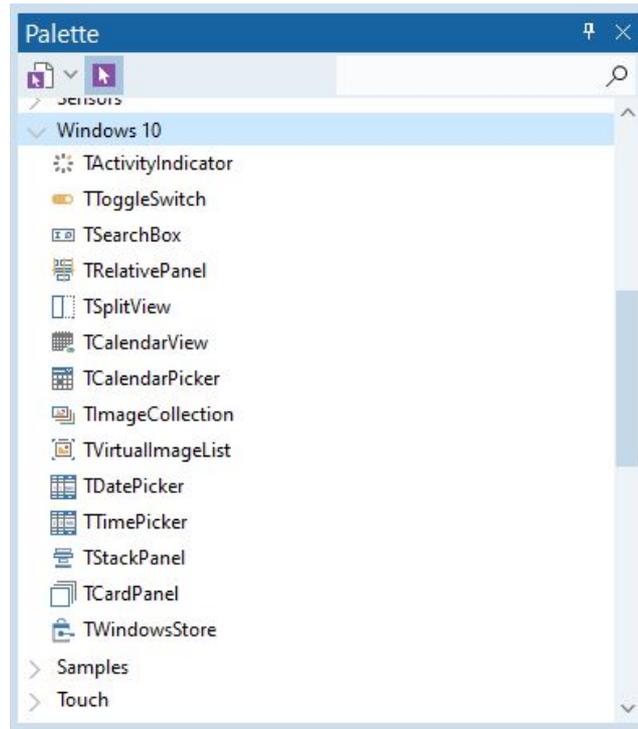
COMPONENTS

- Delphi 10.3 Rio aggiunge il supporto per molti controlli Windows 10, stili e servizi WinRT/UWP
- Estensioni al supporto multi-monitor
- Supporto HiDPI (PerMonitorV2)



COMPONENTS

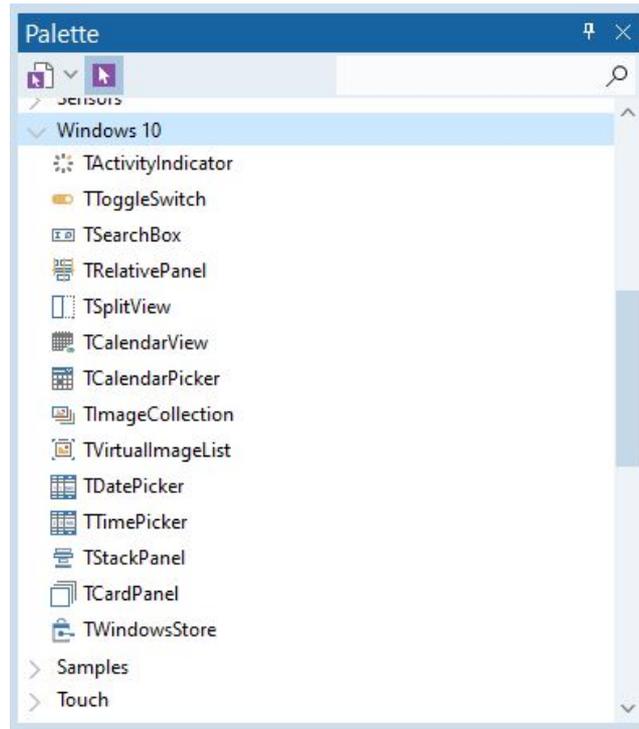
- CalendarView
- CalendarPicker
- SearchBox
- ActivityIndicator
- ToggleSwitch



LAYOUT

Nelle ultime versioni di Delphi sono stati introdotti diversi layout manager:

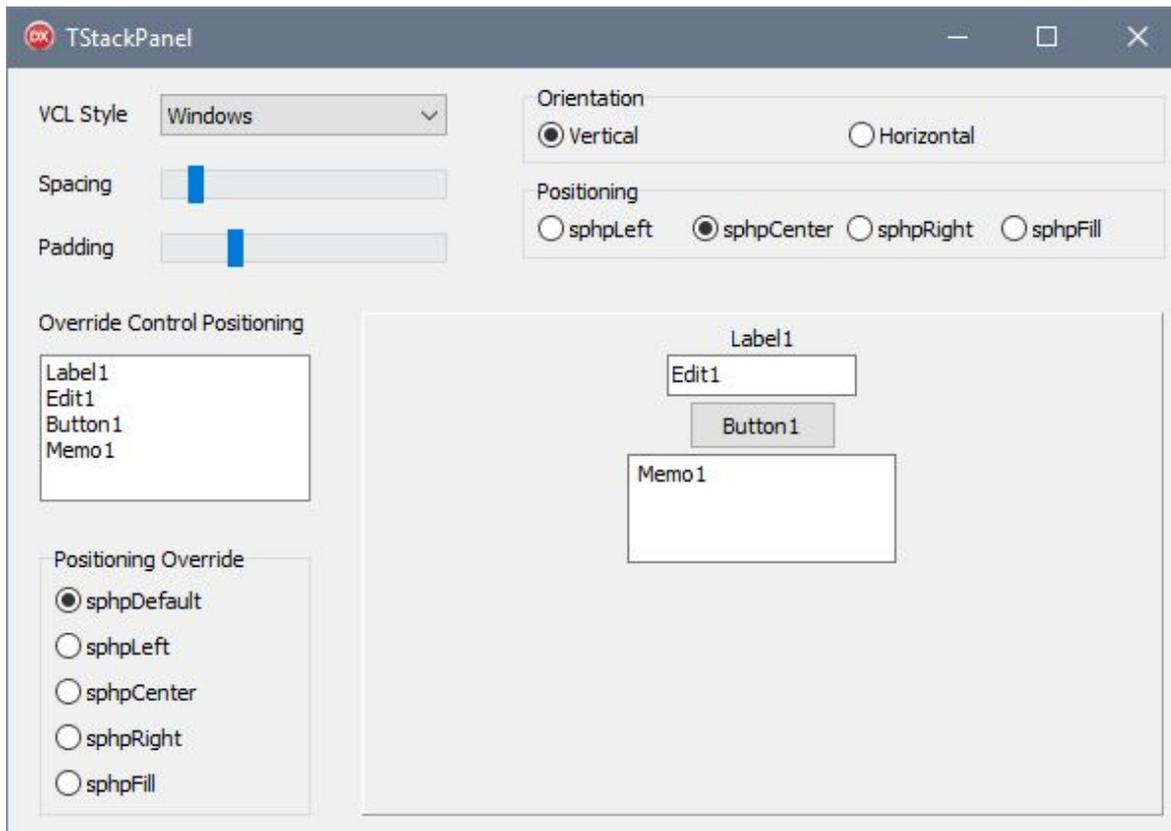
- SplitView
- RelativePanel
- CardPanel
- StackPanel



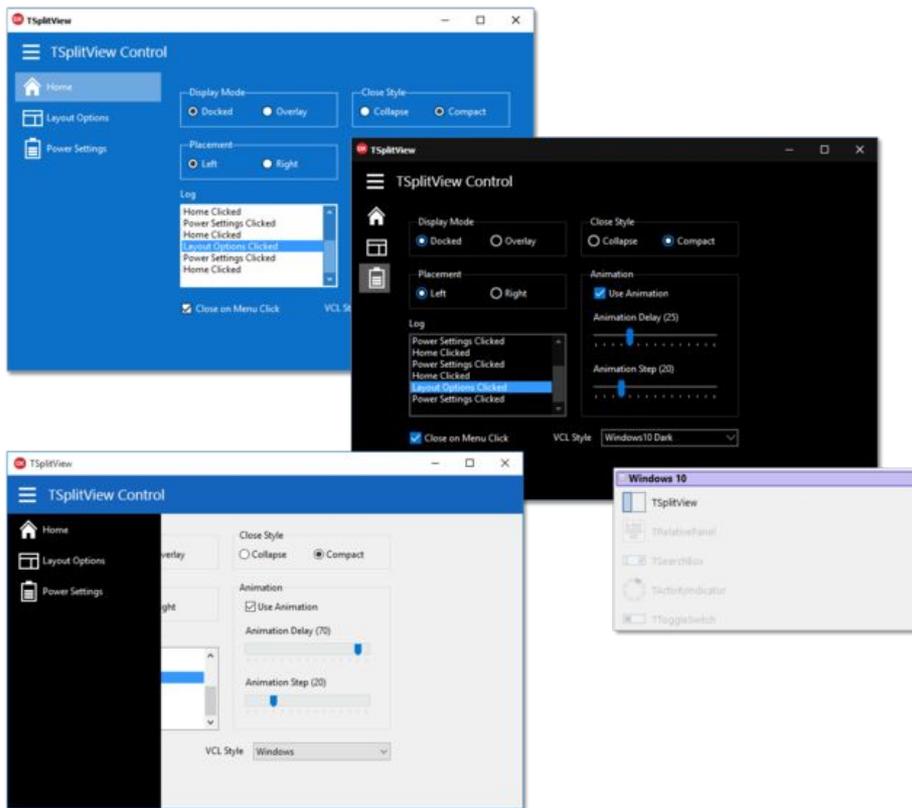
LAYOUT



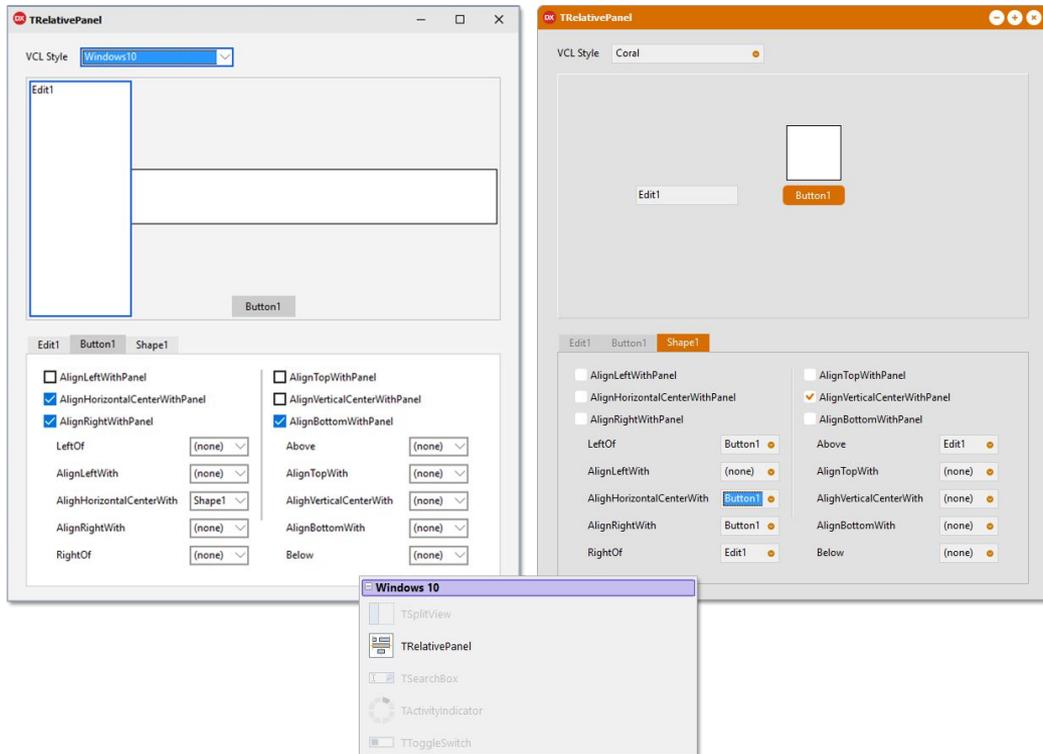
LAYOUT



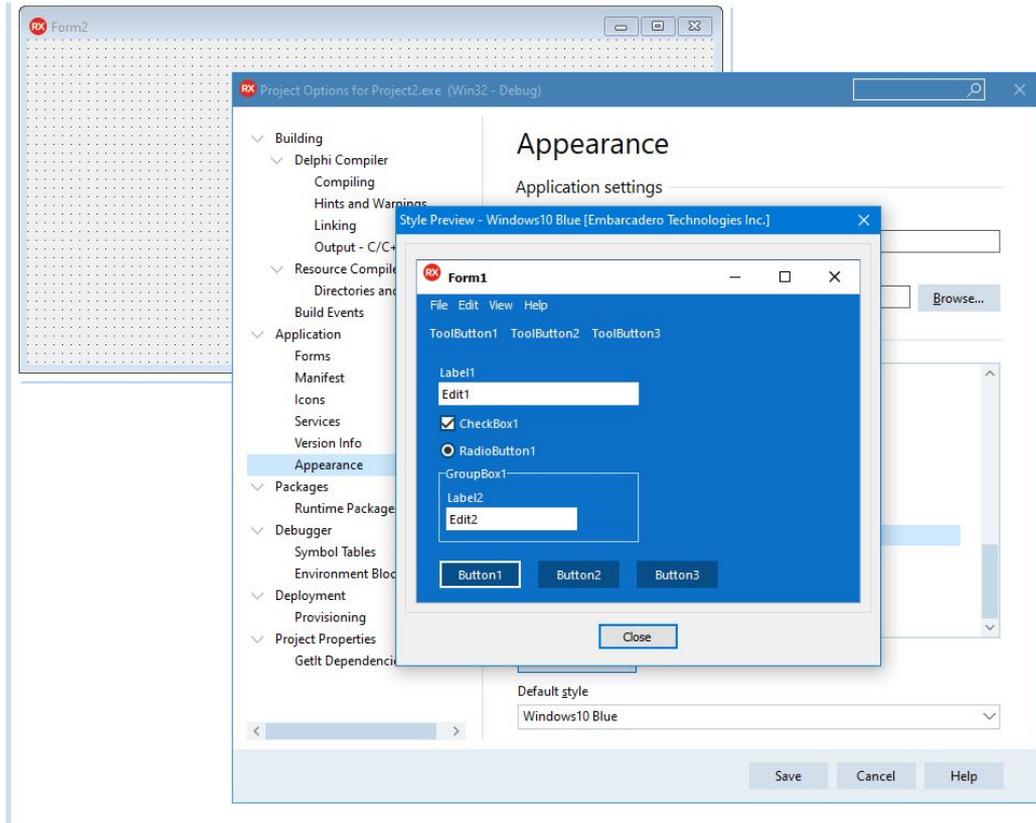
LAYOUT



LAYOUT



STILI



HI-DPI

DPI Awareness

Mode	Windows version	Behavior
Unaware	N/A	All displays are 96 DPI (stretching)
System	Vista	All displays have the same DPI (primary display at login)
Per-Monitor	8.1	Top-level HWND is notified
Per-MonitorV2	Windows 10 Creators Update (1703)	Automatic DPI scaling of comctl32, dialogs...

HI-DPI

- **Compatibilità col passato**
 - La nuova API non è compatibile con le vecchie versioni di windows
 - Nella unit `Vcl.Controls` è stata ridefinita `GetSystemMetrics`
 - Delphi usa la versione più recente che trova
- **Il modello MDI non è più supportato da Windows**

HI-DPI

- Anche usando PerMonitorV2 le immagini vengono scalate (e sfocate)
- Ci sono dei nuovi componenti per risolvere il problema:
 - TImageCollection
 - TVirtualImageList

demo time



Delphi/WinRT

WINAPI

- Usate da Windows sin dalla prima versione
- Costantemente aggiornate
 - Windows 1.0 ne contava ~450 oggi sono migliaia
- Usabili tramite Windows SDK
- In Delphi ci sono le unit per molte di esse ma non per tutte
- Esistono implementazioni non MS: WiNE

WINAPI

- **Calling convention**
 - register, pascal, cdecl, stdcall, safecall
- **Type system**
- **DLL Loading**
 - external, delay loading, LoadLibrary
- **Memory management**
 - Array, string

demo time



COM

- **Interfaccia per componenti software introdotta da Microsoft nel 1993**
 - **DDL (1987):** per scambiare dati tra applicazioni
 - **OLE (1991)** evoluzione di DDL per condividere oggetti ma soprattutto documenti
 - **COM (1993):** comunicazione tra processi
 - **DCOM (1996):** permette il dialogo tra macchine diverse; nasce come risposta a CORBA

COM

- Permette la creazione di componenti software binari
- Scritto in C++ ma funziona in qualsiasi linguaggio
- Questo grazie ad un typesystem condiviso
- Ogni oggetto è unico nel sistema e individuabile attraverso un GUID

COM

- **Gli oggetti COM sono esposti tramite un interfaccia**
- **Tutti gli oggetti devono esporre l'interfaccia IUnknown**
 - **AddRef, Release, QueryInterface**
- **Interfacce predefinite:**
 - **IStream, IOleObject, IDispatch...**

Creare gli oggetti

// Tramite funzioni progettate specificatamente

```
function CreateShape(out Shape: IDrawable): HRESULT;
```

// Tramite funzione generica di creazione

```
function CoCreateInstance(  
    const clsid: TCLSID; // ID della classe  
    unkOuter: IUnknown; // object reuse mechanism  
    dwClsContext: Longint; // in-process, out-process, remote, best  
    const iid: TIID; // ID dell'interfaccia  
    out pv) // puntatore al componente  
: HRESULT; // risultato
```

demo time



WINRT: COS'È

- Nuova API orientata agli oggetti
- Scritta in C++, codice nativo
- Si appoggia a COM e WIN API
- Indipendente dal linguaggio (tramite projections)
 - C++, JavaScript, C#, Delphi...
- Gira in una sandbox
 - Per alcune operazioni è richiesto l'intervento dell'utente

COME SI USA

- La definizione delle API è contenuta in file `.winmd` secondo lo standard ECMA 335
- Per C++ esiste C++/WinRT
 - Nasconde i dettagli implementativi
 - Permette di scrivere codice più “C++”
 - Contiene `cppwinrt.exe` che permette di trasformare i file winmd in header file

API ASINCRONE

- Tutte le API “lente” (>50 millisecondi) sono asincrone:
 - dialog box, accesso ai file, connessione a internet, sockets, streams, dispositivi e servizi, calendario, contatti e appuntamenti...
 - I metodi asincroni hanno il suffisso Async (<Verb>[<Noun>]Async)
 - async/await in C#, helper per C++

WINRT

	Win32	COM	.Net	WinRT
Code Type	Native	Native	Managed	Native
Object Oriented	No	Yes	Yes	Yes
API Definition	None	Type Library	Metadata	Metadata
Non-Blocking	No	Yes	Yes	Yes

Esempio

var

```
Uri: IUriRuntimeClass;  
HttpClient: Http_IHttpClient;
```

begin

```
Uri := Winapi.Foundation.TUri.CreateUri(  
    CreateWinRTStr('http://www.google.com'));
```

```
HttpClient := THttp_HttpClient.Create();
```

```
Progress := HttpClient.GetAsync(Uri);
```

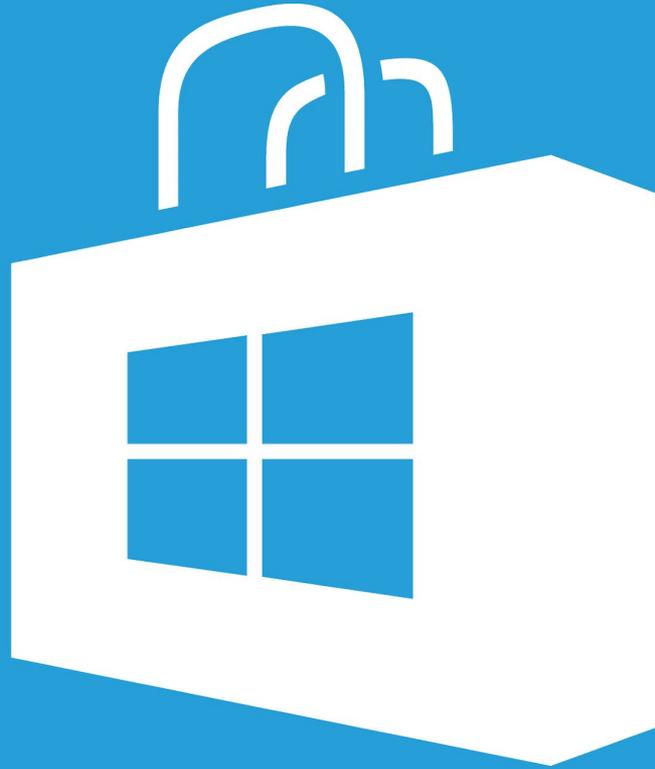
```
...
```

end;

demo time



WINDOWS STORE



WINDOWS STORE

- Piattaforma di distribuzione digitale simile a Google Play e App Store
- Nata con Windows 8 Consumer Preview nel 2012
- Nel 2013 si sono superate le 100.000 App!
- Aggiornamento automatico delle App con l'arrivo di Windows 8.1

MICROSOFT STORE

- Nel 2014, integrazione con Windows Phone Store
- 17 ottobre 2017 (release Fall Creators Update), cambia nome in Microso🖊 Store



MOBILE

- Il 16 dicembre è prevista la chiusura del Windows 8.1 Phone Store
- Windows 8.1 Phone non è più supportato dall'11 luglio 2017
- Windows 10 Mobile sarà supportato fino alla fine del 2019

UWP



UWP 1/2

- **Sicura.** È richiesta l'autorizzazione per l'accesso alle risorse
- **API comuni a tutti i dispositivi (con Windows 10)**
- **Possibilità di utilizzare funzioni specifiche del dispositivo**
- **Responsive!**

UWP 2/2

- **Installazione e disinstallazione più sicura**
- **Linguaggi di programmazione supportati:
C#, C++, Visual Basic, Javascript...**
- **Interfaccia utente con XAML, HTML,
DirectX**

XAML

- eXtensible Application Markup Language
- Nato per l'interfaccia delle applicazioni realizzate con Windows Presentation Foundation (WPF)
- Si basa su oggetti presenti su CLR



E IN DELPHI?

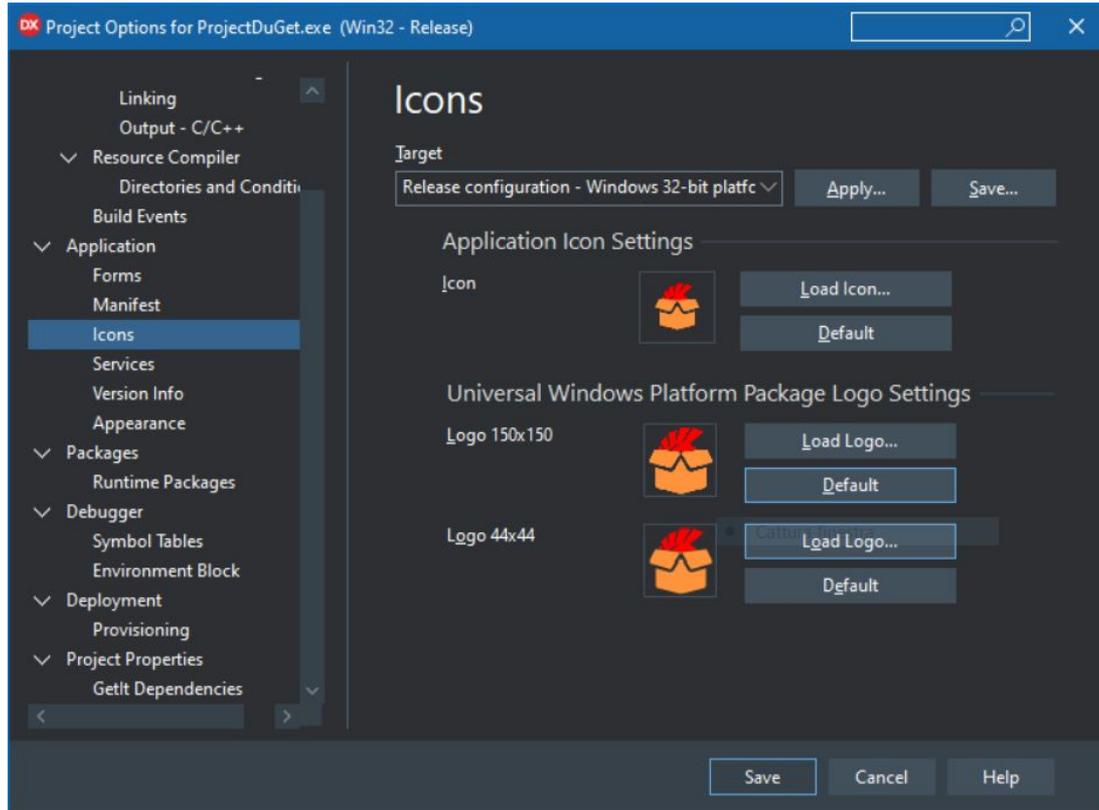
APP DELPHI 1/4

- Supportare localizzazione
- Informativa sulla privacy
- Licenza, se aggiuntiva rispetto alla Standard Application License (*)

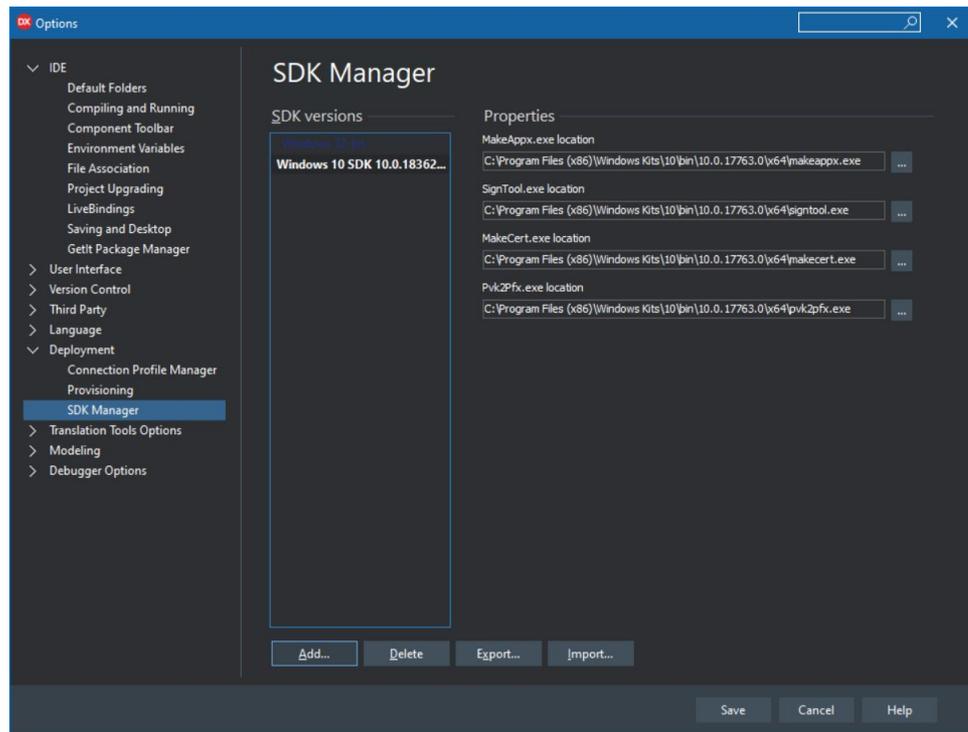
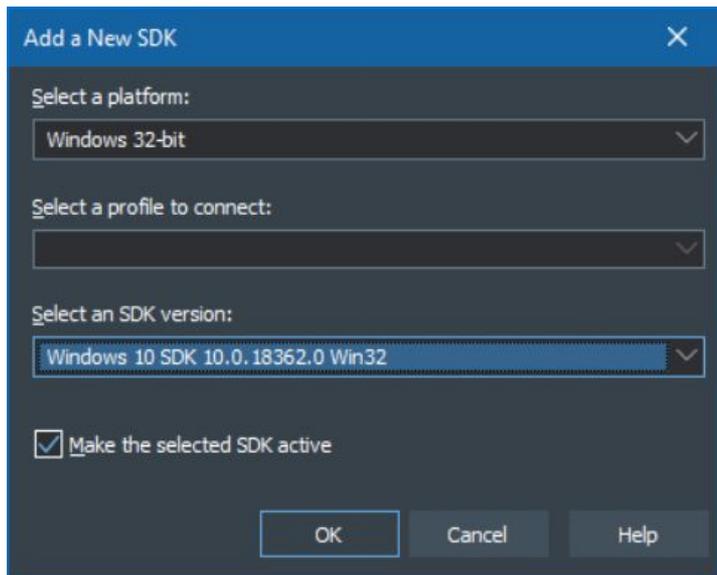


(*) <https://www.microsoft.com/en-MT/store/standard-application-license-terms-eea>

APP DELPHI 2/4



APP DELPHI 3/4



APP DELPHI 4/4

- Deployment di file e risorse aggiuntive
- Creazione pacchetto appx “ad hoc”
- Creazione pacchetto appx per lo Store
- Pubblicazione sul Microso^l Store

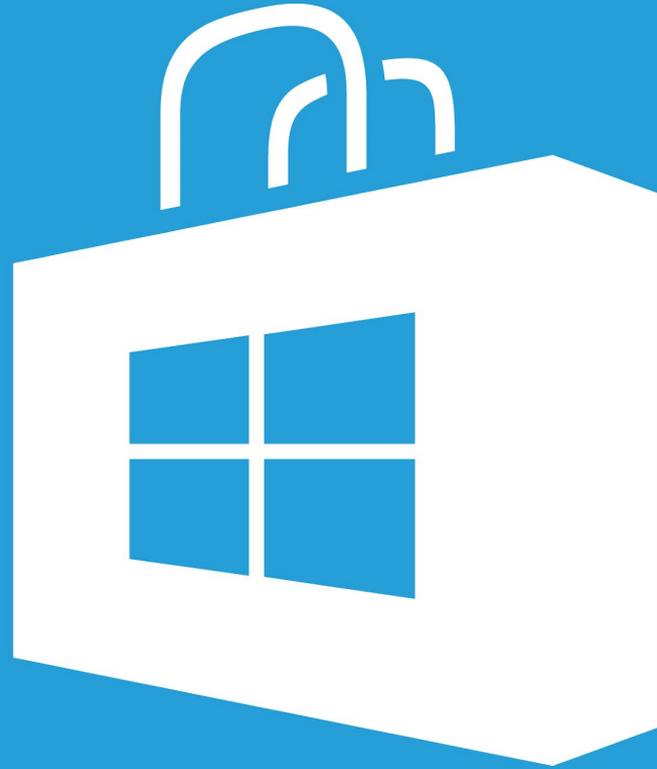
demo time



MSIX

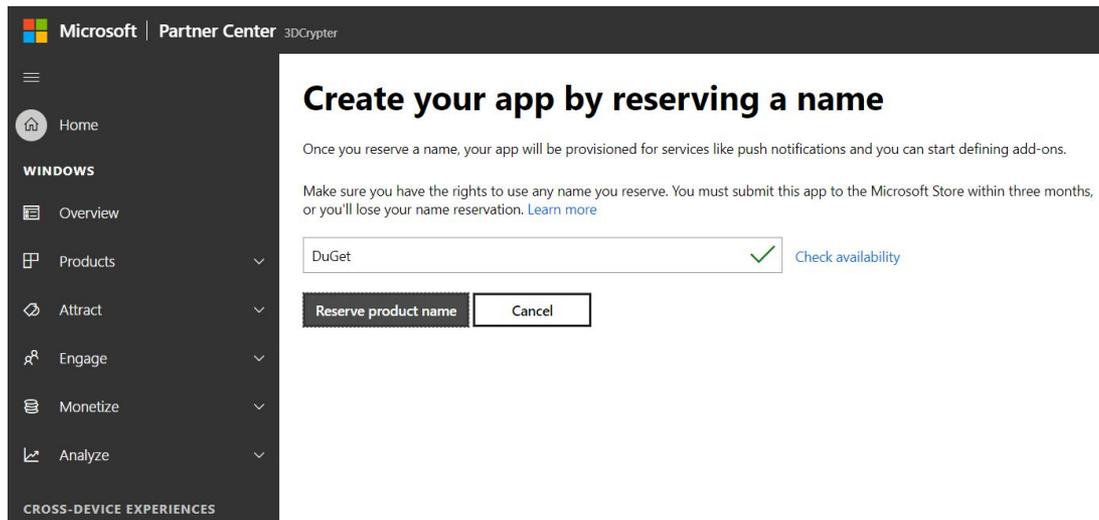
- Nuovo formato per i pacchetti dello Store
- Valido per tutte le Windows apps (UWP, WPF, Win32, WinForm)
- Disponibile un tool di conversione (MSIX Packaging Tool)
- MSIX SDK open-source multipiattaforma

PUBBLICAZIONE



REGISTRAZIONE

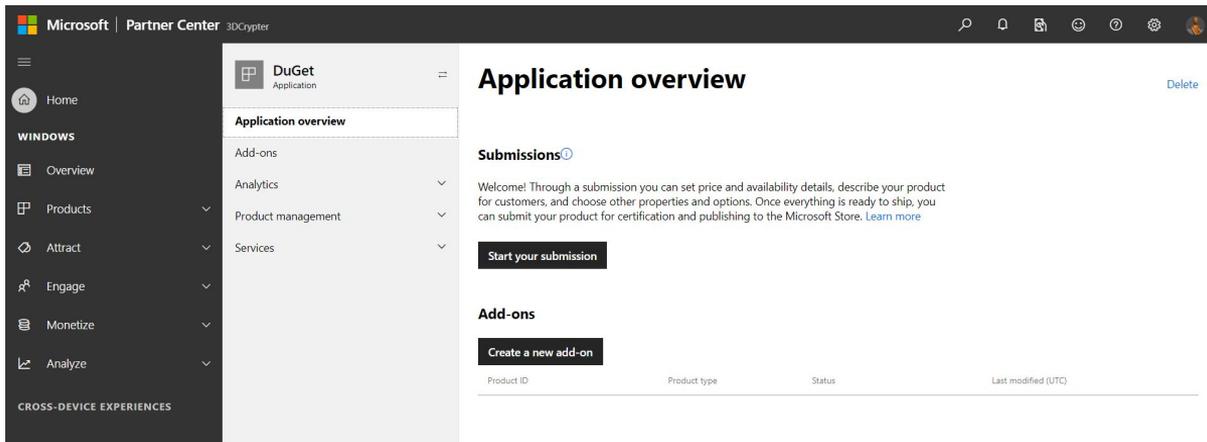
→ Registrare il nome del centro dei Partners
<https://partner.microsoft.com/dashboard>



The screenshot shows the Microsoft Partner Center interface. The top navigation bar includes the Microsoft logo, 'Partner Center', and the user name '3DCrypter'. A left-hand navigation menu lists 'Home', 'WINDOWS' (with sub-items: Overview, Products, Attract, Engage, Monetize, Analyze), and 'CROSS-DEVICE EXPERIENCES'. The main content area is titled 'Create your app by reserving a name'. Below the title, there is explanatory text: 'Once you reserve a name, your app will be provisioned for services like push notifications and you can start defining add-ons. Make sure you have the rights to use any name you reserve. You must submit this app to the Microsoft Store within three months, or you'll lose your name reservation. [Learn more](#)'. A text input field contains 'DuGet' with a green checkmark icon and a 'Check availability' link. Below the input field are two buttons: 'Reserve product name' and 'Cancel'.

SUBMISSION

→ Primo caricamento



→ Compilazione dei vari moduli

FIX

→ Correggere gli errori segnalati e fare attenzione ai warning

ProjectDuGet.appx 3.4 MB

⚠ Package acceptance validation warning: The following restricted capabilities require approval before you can use them in your app: runFullTrust.

Device family availability

This table shows which packages will be offered to specific Windows 10 device families (and earlier OS versions, if applicable) in ranked order. If a device family's box is unchecked, new customers on that type of device won't be able to acquire the app (though customers who already have the app can still use it, and will get any updates you submit).

Let Microsoft decide whether to make this app available to any future device families

Packages	<input checked="" type="checkbox"/> Windows 10 Desktop	<input type="checkbox"/> Windows 10 Mobile	<input type="checkbox"/> Windows 10 Xbox	<input type="checkbox"/> Windows 10 Team	<input type="checkbox"/> Windows 10 Holographic	Windows 8/8.1	Windows Phone 8.x and earlier
ProjectDuGet.appx v1.0.0.0, X86	1						

RUNFULLTRUST?

- Normale per applicazioni desktop
- `runFullTrust`: esecuzione con massimo livello dei permessi
- L'App verrà sottoposta a controlli più approfonditi da parte del revisore



MA PRIMA...

- Le app vengono testate su un ambiente Windows 10 S
- Assicurarsi che l'applicazione funzioni su Windows 10 S prima di pubblicarla



TEST 1/2

1. Scaricare lo zip indicato nella documentazione:
<https://docs.microsoft.com/it-it/windows/msix/desktop/desktop-to-uwp-test-windows-s>
2. Preparare una VM con la policy Audit e installare il certificato AppxTestRootAgency.cer
3. Creare una configurazione su Delphi con i dati del certificato AppxTestRootAgency.pfx
4. Eseguire la creazione del pacchetto appx... per avere il deployment

TEST 2/2

5. Creare il pacchetto firmato a mano:

```
makeappx pack -p "APPX PATH" -d "DEPLOYMENT DATA PATH" -l  
signtool.exe sign /a /v /fd SHA256 /f "AppxTestRootAgency.pfx" "APPX PATH"
```

6. Installare l'appx sulla macchina virtuale per il test

Consiglio: installare PAServer sulla macchina virtuale e usare il debug remoto

 **DelphiDay**
italian conference

PADOVA
EDITION

Edizione 2019

THANK YOU