



توثيق Nexamas.UI

دليل ثنائي اللغة لتقييم Community Preview وبناء التطبيقات وفهم حدود
.Professional

عقد Community Preview

1.0.0-preview.1

يصف هذا التوثيق عقد الإصدار التجريبي preview.1-1.0.0 المعتمد. يبقى تنزيل الحزمة العامة معطلاً حتى
.Website Phase 40.9

المصدر: عقد محتوى Website Phase 40.4 الموحد

آخر تحديث: 28-07-2026

© Nexamas. All rights reserved 2026

المحتويات

01	نظرة عامة	10	واجهات البيانات
02	متطلبات النظام	11	دليل Showcase
03	التثبيت عبر NuGet و ZIP	12	ترخيص Community
04	بداية سريعة	13	مقارنة الإصدار المجاني والاحترافي
05	قائمة أول تطبيق	14	إدارة الإصدارات
06	هيكل التطبيق	15	استكشاف الأخطاء
07	البيئات والأسطح	16	ملاحظات الإصدار
08	نظرة على عناصر التحكم	17	سياسة الدعم
09	التخطيط والأحجام		

Nexamas.UI هي حزمة SDK مغلقة المصدر لواجهات Windows Forms، وتقدم نموذج MAS API عامًا واحدًا على .NET Framework 4.8 و .NET 10 for Windows.

اسم المنتج وحزمة NuGet والتجميعية وجذر مساحات الأسماء هو Nexamas.UI. تستخدم الواجهة العامة للمستهلك عائلة MAS* المستقرة، ومنها MASApplicationWindow و MASApplication و MASButton و MASDataGrid والأنواع المرتبطة بها.

- تكوين التطبيق والنوافذ عبر واجهات التطبيق العامة.
- مصانع Controls typed، وبناء Layout، والقيمات والمواد، وأساس الترجمة وRTL.
- واجهات عرض البيانات وسير العمل، مع بنية قياس الجودة والتحقق البصري.
- نموذج منتج موحد على كلا إطار العمل المدعومين.

حدود المنتج

Nexamas.UI هي SDK لسطح مكتب Windows. لا يدعي هذا الإصدار دعم macOS أو Linux أو الهاتف أو المتصفح أو تصدير PDF العام أو التكامل مع طابعة نظام التشغيل.

مسار الإثبات الموثق هو Windows Desktop وWindows Forms وبنية x64 | Release مع إطار عمل مدعومين.

المتطلب	العقد المدعوم
نظام التشغيل	Windows Desktop
مضيف الواجهة	Windows Forms
إطارات العمل	net10.0-windows10.0.19041.0 و net48
مسار الإصدار الموثق	Release x64
لغات المستهلك	Visual Basic و #C
SDK الحديث	NET 10 SDK.
البناء الكلاسيكي	Visual Studio/MSBuild مع دعم استهداف .NET Framework 4.8

نطاق الإصدار

لا تعرض x86 أو AnyCPU كأهداف إصدار تجاري معتمدة ما لم يصفها إثبات لاحق صراحة.

يستخدم الإصدار التجريبي هوية الحزمة Nexamas.UI ورقم 1.0.0-1-preview.1. يبقى التوزيع العام محجوبًا عمدًا حتى Phase 40.9.

بوابة نشر Preview

تصف الأوامر أدناه عقد الحزمة المعتمد، لكنها لن تصبح قابلة للاستخدام العام إلا بعد نجاح ملفات الإصدار والترخيص والصفحات القانونية والبصمات وبوابات التحقق النهائية.

ملف مشروع .NET 10

xml

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>WinExe</OutputType>
    <TargetFramework>net10.0-windows10.0.19041.0</TargetFramework>
    <UseWindowsForms>true</UseWindowsForms>
    <EnableWindowsTargeting>true</EnableWindowsTargeting>
    <PlatformTarget>x64</PlatformTarget>
    <Prefer32Bit>>false</Prefer32Bit>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Nexamas.UI" Version="1.0.0-preview.1" />
  </ItemGroup>
</Project>
```

محلي أو خاص Feed إعداد

xml

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <clear />
    <add key="NexamasPreview" value="PATH_OR_FEED_PROVIDED_BY_NEXAMAS" />
    <add key="NuGetOfficial" value="https://api.nuget.org/v3/index.json" />
  </packageSources>
</configuration>
```

- يجب أن يحتوي ZIP على الحزمة المعتمدة نفسها، والترخيص، وملاحظات الإصدار، وبيان SHA-256.
- لا تنسخ التجميعات يدويًا بين مجلدات إطارات العمل عندما تكون NuGet متاحة.
- يمكن حفظ Community في مستودع داخلي خاص بالشركة، لكن لا يجوز نسخها إلى Feed عام.

أنشئ تطبيق Windows Forms، وأضف مرجع Nexamas.UI، ثم اربط أو أنشئ نافذة MAS.

Visual Basic أول نافذة بـ

vb

```
Option Strict On
Option Explicit On

Imports System
Imports System.Windows.Forms
Imports Nexamas.UI.Application
Imports Nexamas.UI.Controls
Imports Nexamas.UI.Layout

Public NotInheritable Class MainForm
    Inherits Form

    Private ReadOnly _window As MASApplicationWindow

    Public Sub New()
        Text = "Nexamas UI Quickstart"
        Width = 1000
        Height = 700
        StartPosition = FormStartPosition.CenterScreen

        _window = MASApplication.AttachWindow(
            owner:=Me,
            configure:=Sub(window As MASApplicationWindow)
                Dim title As MASTitle = window.Controls.AddTitle("Nexamas UI")
                Dim message As MASLabel = window.Controls.AddLabel("One MAS API on net48 and .NET 10.")
                Dim action As MASButton = window.Controls.AddButton("Ready")

                AddHandler action.Click,
                    Sub(sender As Object, e As EventArgs)
                        window.Services.Toasts.Success("The package consumer is alive.", "Quickstart")
                    End Sub

                window.Controls.LayoutPage(
                    Sub(page As MASApplicationLayoutPageBuilder)
                        page.Padding(MASLayoutSpacing.Spacious)
                        page.Spacing(MASLayoutSpacing.Medium)
                        page.FullWidth(title, MASSize.FillWidth)
                        page.FullWidth(message, MASSize.FillWidth)
                        page.ActionGroup(
                            Sub(actions As MASApplicationActionGroupLayoutBuilder)
                                actions.AlignCenter().EqualItemWidth().Add(action, MASSize.Default)
                            End Sub)
                    End Sub)
            End Sub)
    End Sub

    Protected Overrides Sub OnFormClosed(e As FormClosedEventArgs)
        If _window IsNot Nothing Then _window.Dispose()
        MyBase.OnFormClosed(e)
    End Sub
End Class
```

```
using System;
using System.Windows.Forms;
using Nexamas.UI.Application;
using Nexamas.UI.Controls;
using Nexamas.UI.Layout;

internal sealed class MainForm : Form
{
    private readonly MASApplication _application;
    private readonly MASApplicationWindow _window;

    internal MainForm()
    {
        Text = "Nexamas UI Quickstart";
        Width = 1000;
        Height = 700;

        _application = MASApplication.Create();
        _window = _application.CreateWindow(this);

        MASTitle title = _window.Controls.AddTitle("Nexamas UI");
        MASLabel message = _window.Controls.AddLabel("One MAS API on net48 and .NET 10.");
        MASButton action = _window.Controls.AddButton("Ready");

        _window.Controls.LayoutPage(page =>
        {
            page.Padding(MASLayoutSpacing.Spacious);
            page.Spacing(MASLayoutSpacing.Medium);
            page.FullWidth(title, MASSize.FillWidth);
            page.FullWidth(message, MASSize.FillWidth);
            page.ActionGroup(actions => actions.AlignCenter().EqualItemWidth()).Add(action,
            MASSize.Default));
        });
    }

    protected override void OnFormClosed(FormClosedEventArgs e)
    {
        _window?.Dispose();
        _application?.Dispose();
        base.OnFormClosed(e);
    }
}
```

قاعدة دورة الحياة

تخلص من كل `MASApplicationWindow`. وعند استخدام `MASApplication.Create()` تخلص من التطبيق بعد إغلاق جميع النوافذ التي يملكها.

ابدأ بشريحة صغيرة قبل نقل شاشة إنتاجية: نافذة واحدة، عنوان، Label، إجراء واحد وLayout بسيط.

- اختر net48 أو net10.0-windows10.0.19041.0 وبنية x64.
- أضف الإصدار التجريبي المحدد من Feed أو ZIP المعتمد.
- أنشئ Form عاديًا في WinForms واربط MASApplicationWindow.
- أضف عناصر التحكم عبر مصانع window.Controls typed.
- كوّن الصفحة عبر LayoutPage وMASSize بدل تثبيت المواقع بالعكس.
- تخلص من النافذة وواجهة التطبيق بصورة صحيحة.
- اختبر تركيز لوحة المفاتيح وDPI وتغيير الحجم واتجاه الترجمة وبيانات واقعية.

نصيحة للتقييم

قيّم Nexamas.UI عبر نقل سير عمل واقعي واحد، وليس عبر الحكم على زر منفرد.

صُممت طبقة التطبيق العامة لتنسيق النوافذ والتنقل والتخطيط والأوامر والـ Overlays والتغذية الراجعة والخدمات التي يملكها التطبيق.

استخدم `MASApplication.AttachWindow(...)` لتطبيق صغير بنافذة واحدة. واستخدم `MASApplication.Create()` و `CreateWindow(...)` عندما تملك نسخة تطبيق واحدة عدة نوافذ.

مخطط تكوين التطبيق

vb

```
Dim application As MASApplication = MASApplication.Create()
Dim window As MASApplicationWindow = application.CreateWindow(owner:=Me)

Dim navigation = window.Controls.AddNavigationRail()
Dim title = window.Controls.AddTitle("Operations")
Dim content = window.Controls.AddPanel()

window.Controls.LayoutPage(
    Sub(page)
        page.Padding(MASLayoutSpacing.Spacious)
        page.Spacing(MASLayoutSpacing.Medium)
        page.FullWidth(title, MASSize.FillWidth)
        page.FullWidth(content, MASSize.Fill)
    End Sub)
```

- اجعل دورة حياة النوافذ صريحة.
- استخدم بوابات الخدمات التي يملكها التطبيق بدل Global singletons.
- عامل تكوين Showcase كمثال، لا كمصدر سلطة للادعاءات العامة.
- لا تستدع أنظمة Friend/internal مباشرة من تطبيقات العملاء.

توفر Nexamas.UI بوابات عامة للتيّمات والأسطح، بينما تبقى بنية التصميم والتحقق والاستوديو الأعمق داخلية ما لم تُكشف بعقد مستقل.

حد الثيم العام

vb

```
Dim app As MASApplication = MASApplication.Create()
Dim window As MASApplicationWindow = app.CreateWindow(owner:=Me)

' Use the public application/window theme gateway.
' Keep theme selection at the application boundary rather than styling each control independently.
Dim themeGateway = window.Theme
```

- اختر الثيم على مستوى التطبيق أو النافذة.
- استخدم وصفات مشتركة للمسافات والأحجام والمواد والحالات للحفاظ على الاتساق.
- تحقق من التباين وحالات التركيز في الثيمين الفاتح والداكن.
- لا يُدعى أن Theme Studio و Surface Designer واجهات SDK عامة في هذا الإصدار.

يحتوي المخزون المعتمد على 80 عنصر تحكم مصنّفًا بين Community و Community + Pro ومسارات Professional والمشاركة.

Professional

20

عنصرًا متقدمًا

Community + Pro

9

عناصر مشتركة

Community

51

عنصرًا أساسيًا

- التطبيق والتنقل: العناوين والأسطح الأمرية والتنقل واللوحات والـ Overlays والتغذية الراجعة.
- الإدخال والاختيار: الأزرار والنصوص والمحددات والقوائم والأشجار والمحركات وسلوك التركيز.
- البيانات وسير العمل: Views و Grids و Master-detail والتخطيط ولوحات المعلومات والفلاتر والرسوم والملفات.
- تُباع عناصر Professional المتقدمة حسب سياسة الإصدارات فقط عندما يكون المصدر والـ Public API والإثبات موجودًا.

المخزون ليس اختصارًا للجاهزية

تصنيف عنصر تحكم لا يعني تلقائيًا أن كل سيناريو أو نطاق أداء أو مسار تصميم جاهز للإنتاج. استخدم ملاحظات الإصدار وأدلة المشكلات المعروفة للبناء المحدد.

استخدم بناء Layout العامة وMASSize وMASLayoutSpacing للتعبير عن المقصود بدل تحديد موقع كل عنصر يدويًا.

مثال تخطيط صفحة

vb

```

window.Controls.LayoutPage(
    Sub(page As MASApplicationLayoutPageBuilder)
        page.Padding(MASLayoutSpacing.Spacious)
        page.Spacing(MASLayoutSpacing.Medium)
        page.FullWidth(header, MASSize.FillWidth)
        page.FullWidth(dataGrid, MASSize.Fill)
        page.ActionGroup(
            Sub(actions)
                actions.AlignEnd().EqualItemWidth()
                actions.Add(cancelButton, MASSize.Default)
                actions.Add(saveButton, MASSize.Default)
            End Sub)
    End Sub)

```

- استخدم FillWidth أو Fill للأسطح التي يجب أن تتغير مع حجم النافذة.
- استخدم Action Groups لصفوف الأوامر المحاذاة.
- اختبر الأحجام الدنيا و DPI والنصوص المترجمة الطويلة واتجاه RTL.
- لا تعتمد على أصناف responsive أو virtualization الداخلية كواجهات عامة.

يملك DataGridView و DataView واجهات عامة موثقة للمستهلك، بينما تخضع عناصر سير العمل الاحترافية الأخرى لمخزون الإصدارات المعتمد.

نقاط دخول واجهات البيانات

vb

```
Dim grid As MASDataGrid = window.Controls.AddDataGrid()  
Dim view As MASDataView = window.Controls.AddDataView()  
  
' Bind through the documented public consumer surface for your release.  
' Validate large-data behavior with representative customer data before shipping.
```

- ابدأ بعدد صفوف واقعي وأشكال أعمدة ممثلة لحالتك.
- تحقق من الاختيار والتحرير والتنقل بلوحة المفاتيح والفرز والتصفية وتغيير الحجم المطلوب لسير العمل.
- عامل محركات virtualization الداخلية كتفاصيل تنفيذ ما لم تكشف بوابة عامة عقدًا صريحًا.
- لا يجوز نسخ عناصر Professional أو الوصول إليها بالانعكاس أو استدعاؤها من Community عبر واجهات داخلية.

يعرض Showcase تكوينات ورحلات عميل مثبتة، لكنه لا يعيد تعريف حقيقة الحزمة أو الإصدار أو الترخيص أو API.

- استخدم صورًا ومصدر مثال يستهلك واجهات Nexamas.UI العامة فقط.
- يمكن نشر تطبيق المثال لأنه يستهلك SDK، بينما يبقى مصدر SDK المغلق خاصًا.
- يجب أن تعمل عروض Professional عبر الحدود البعيدة أو المرخصة المعتمدة، وألا ترسل ملفات Nexamas.UI.Pro إلى المتصفح.
- تحقق من المثال على Windows وWindows Forms وx64 وإطار العمل الموثق قبل ربطه بإصدار.

حد المرحلة

يُستكمل المثال القابل للتشغيل ومجموعة الصور النهائية وأدلة المتصفح في Website Phase 40.5.

Community مجانية للاستخدام الشخصي والتعليمي وغير التجاري والتجاري وSaaS والأنظمة الداخلية والتطبيقات المدفوعة ومنتجات الاشتراك.

- لا رسوم Runtime ولا مفتاح ترخيص ولا حد للمقاعد أو المستخدمين أو المشاريع.
- يمكن تضمين DLL غير المعدلة داخل تطبيق نهائي يحتاجها.
- يمكن بيع تطبيقات وقوالب وإضافات وعناصر تحكم ومكتبات أصلية تضيف قيمة مستقلة.
- يمكن حفظ نسخة مطابقة وغير معدلة في مستودع خاص بمؤسسة واحدة.
- لا تبع Nexamas.UI منفردة ولا تغير اسمها ولا تعيد تغليفها ولا تنشر Mirror أو Feed عامًا.
- لا تزل إشعارات الملكية ولا تنشئ SDK مشتقة منافسة.
- تُمنع الهندسة العكسية والتعديلات الداخلية إلا حيث يفرض القانون استثناءً.
- لا يُلزم إظهار النسبة. نقدر توصية ودية بالمشروع عندما تساعدك Nexamas.UI.

المراجعة القانونية مطلوبة

تلخص هذه الصفحة سياسة المنتج المعتمدة. يجب أن يمر نص الترخيص العام النهائي بمراجعة قانونية مهنية قبل النشر في Phase 40.9.

مقارنة الإصدار المجاني والاحترافي

13

صُممت Community لتكون مفيدة دون حدود Runtime مصطنعة، بينما تضيف Professional عناصر متقدمة وحقوق تطوير مرخصة ودعمًا مباشرًا وقناة تسليم خاصة.

القدرة	Community	Professional
نموذج التطبيق والنوافذ والبيئات و DPI والإدخال والتخطيط والترجمة و RTL	مضمن	مضمن
عناصر Community الأساسية	51 عنصرًا	كل عناصر Community
العناصر المشتركة ذات المسارات المتقدمة	سلوك أساسي مفيد على 9 عناصر	الأساس مع المسارات المتقدمة المرخصة
عناصر Professional	غير مضمنة	20 عنصرًا للتحليلات وسير العمل ومساحة التطبيق والمحركات والتنقل
إطارات العمل المستهدفة	.NET Framework 4.8 و .NET 10 for Windows	الأهداف نفسها
لغات المستهلك	Visual Basic و C#	Visual Basic و C#
الاستخدام التجاري و SaaS والأنظمة الداخلية	مسموح	مسموح
رسوم Runtime أو مقاعد المستخدم النهائي	لا يوجد	لا يوجد
الحاجة لاتصال مستمر بالإنترنت	لا يوجد	لا يوجد للاستخدام المرخص
الأمثلة ومواد التبنّي	توثيق أساسي ووصفات وأمثلة للمبتدئين	أمثلة متقدمة وقوالب تجارية
التحقق البصري وسير عمل الجودة	بوابة تحقق عامة	خطوط أساس وتقارير وسير عمل فرق متقدم
الدعم الفني المباشر	غير مضمون	مضمن أثناء مدة الدعم والتحديث الفعالة
توزيع الحزم	Prerelease عام مخطط له في Website Phase 40.9	تسليم خاص فقط؛ الإطلاق المدفوع ضمن Commercialization Phase 16

الخطة	السعر	المدة أو الغرض
Community	€0	استخدام بلا حد زمني
Professional سنوية	€399	حقوق تطوير لمدة 12 شهرًا
Professional لثلاث سنوات	€999	حقوق تطوير لمدة 36 شهرًا
Professional دائمة	€1,499	استخدام دائم للإصدارات المؤهلة + 12 شهرًا تحديث ودعم
تجديد التحديثات والدعم	€199	12 شهرًا إضافية من التحديثات والدعم، اختياري
Professional Trial	€0	45 يومًا من التفعيل، للتقييم فقط
Enterprise	عرض مخصص	عرض وشروط مخصصة

نشر Runtime

لا تتضمن الخطط المدفوعة رسوم Runtime أو رسوم مقاعد المستخدم النهائي. تستمر التطبيقات المنشورة أثناء صلاحية رخصة المدة في العمل.

تستخدم Nexamas.UI Semantic Versioning. أول Community Preview عام هو 1.0.0-1.0.0.preview.1، بينما يبقى AssemblyVersion وFileVersion عند 1.0.0.0 لخط الأساس المعتمد.

نوع التغيير	أثره على الإصدار
إصلاح متوافق أو أمني أو تصحيح توثيق	Patch
إضافة عامة متوافقة	Minor
كسر متعمد للواجهة العامة أو السلوك الموثق	Major مع دليل ترحيل واعتماد المالك
بناء تقييم قبل اعتماد Stable	لاحقة Prerelease مثل 1.0.0.preview.1

- تحمل Community وProfessional رقم إصدار المنتج نفسه.
- إصدار الحزمة المنشور غير قابل للتغيير؛ أي Bytes مختلفة تتطلب إصدارًا جديدًا.
- لا تستخدم عبارة Lifetime Updates. الرخصة الدائمة تعني استخدامًا دائمًا للإصدارات المؤهلة، لا تحديثات مستقبلية غير محدودة.
- تحدد أهلية التحديث عبر وقت الإصدار الموثوق واستحقاق العميل.

تحقق أولاً من إطار العمل والمنصة ومصدر الحزمة واستخدام الـ Public API ونمط التلخيص قبل البحث في سلوك أعمق.

العرض	ما يجب فحصه
تعذر استعادة الحزمة	تحقق من مسار Feed/ZIP المعتمد ورقم Prerelease الدقيق وأولوية مصادر NuGet.
بناء NET 10.0 يرفض أنواع WinForms	استخدم net10.0-windows10.0.19041.0 و UseWindowsForms=true و EnableWindowsTargeting=true و x64.
بقاء النافذة أو الخدمة بعد الإغلاق	تخلص من MASApplicationWindow ومن MASApplication المالكة في التطبيقات ذات الدورة الصريحة.
Layout يقص النص المترجم	اختبر الحجم الأدنى و DPI وسلوك Fill/FillWidth والنصوص الطويلة RTL/LTR.
مسار مذكور غير موجود	تحقق من أنه Public في الحزمة نفسها. تصدير PDF والطباعة وعدة Studios داخلية ليست مزايا Preview عامة.

- سجل رقم الحزمة وإطار العمل وإصدار Windows والمعمارية ومثالاً مصغراً يعيد المشكلة.
- بالنسبة إلى Community استخدم التوثيق والتحديثات العامة؛ الدعم المباشر غير مضمون.
- بالنسبة إلى Professional أرفق تفاصيل الاستحقاق دون نشر مادة الترخيص الخاصة علناً.

preview.1-1.0.0 هي هوية أول Community Preview عام معتمد، ولا تعيد كتابة خط الأساس الهندسي الداخلي المغلق 1.0.0.

- نموذج منتج موحد لـ net48 و net10.0-windows10.0.19041.0.
- هوية حزمة Nexamas.UI مستقرة وعائلة MAS * عامة للمستهلك.
- تصنيف معتمد لعناصر Community + Pro و Professional.
- توثيق موقع وملفات PDF بالعربية والإنجليزية من عقد محتوى واحد.
- لا يوجد ملف Professional عام أو Checkout أو تسليم Trial إنتاجي في Website Phase 40.

ليس تنزيلًا عامًا بعد

يُفَعَّل رابط تنزيل Preview فقط في Website Phase 40.9 بعد نجاح بوابات الحزم والقانون والخصوصية و SEO والبصمات والمتصفح واعتماد المالك.

تعتمد Community على التوثيق والتحديثات العامة. تتضمن Professional هدف رد أولي غير تعاقدية، بينما تتطلب خدمات Enterprise اتفاقاً مستقلاً.

الإصدار	حد الدعم
Community	التوثيق وملاحظات الإصدار والتحديثات العامة؛ دون رد مباشر مضمون.
Professional	هدف الرد الأولي: بنهاية يوم عمل Nexamas التالي؛ ليس SLA تعاقدياً ولا يضمن وقت الحل.
Enterprise	يتوفر SLA تعاقدية فقط عندما يُذكر صراحة في نموذج الطلب.

- هدف Professional هو رد أولي بنهاية يوم عمل Nexamas التالي.
- هذا الهدف ليس SLA تعاقدياً ولا يضمن مدة حل المشكلة.
- حقوق الاستخدام الدائم منفصلة عن أهلية التحديث والدعم.
- يضيف التجديد السنوي الاختياري 199€ مدة 12 شهراً من التحديثات والدعم؛ ولا يجدد حق الاستخدام الدائم أو يمدده لأنه لا ينتهي.