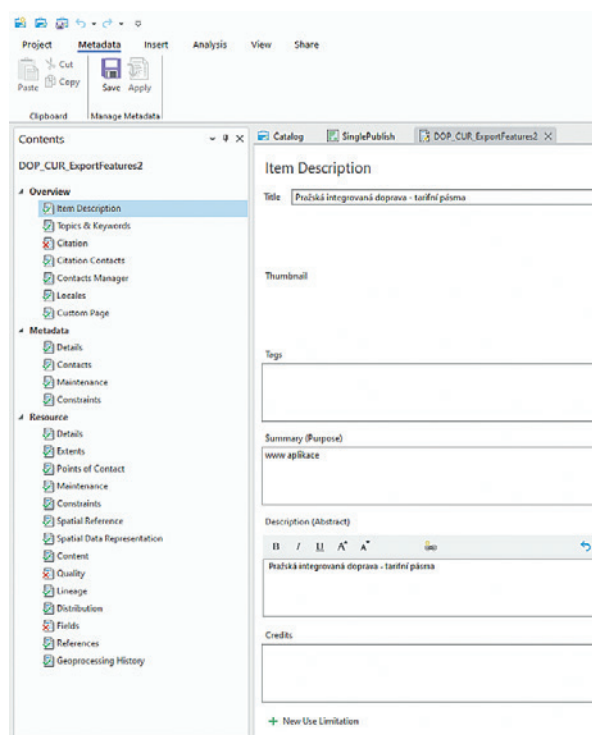


Přizpůsobení metadatového editoru ArcGIS Pro

Róbert Spál, Institut plánování a rozvoje hl. m. Prahy

Potřebovala by vaše organizace upravit metadatový editor ArcGIS Pro tak, aby bylo možné uchovávat si vlastní metadatové položky, které nejsou standardně k dispozici? Pak jsou následující řádky určené právě vám.

Pro správu a editaci metadat využíval *Institut plánování a rozvoje hlavního města Prahy* (dále jen IPR) dlouhé roky metadatový editor ArcMap. Vzhledem k pomalému ale jistému zániku ArcMapu však vyvstala nutnost přemigrovat metadata do prostředí ArcGIS Pro.



Obr. 1. Metadatový editor ArcGIS Pro.

Jelikož má IPR několik vlastních metadatových položek, které nejsou standardní součástí žádného z metadatových profilů ArcGIS Pro, bylo nutné tyto změny doprogramovat a vytvořit doplněk pro ArcGIS Pro, který by tyto

potřeby reflektoval. Následující odstavce se vám proto pokusí přiblížit, jak takové změny můžeme dosáhnout a na co vše si dát pozor, když potřebujeme přidat vlastní metadatová pole do ArcGIS Pro. Jelikož IPR musí mít metadata v souladu s českou lokalizací INSPIRE, tak zde nakousneme také řešení problému, kdy bylo nutné upravit příslušný soubor XSLT tak, aby naše metadata mohla být validována oproti českému INSPIRE.

PŘÍPRAVA, INSTALACE A NASTAVENÍ

Pro začátek je nutné stáhnout a nainstalovat **metadata toolkit**, který naleznete na GitHubu Esri¹. Je to vlastně souhrn různých souborů, pomocí kterých dokážeme upravit předmětný editor v ArcGIS Pro vytvořením vlastního add-inu (doplňku). Na stejné stránce naleznete i podrobný návod, jak krok za krokem postupovat. Po stažení a instalaci je také nutné správně nastavit celý projekt, s čímž vám opět pomůže návod².

Důležité je vědět, že pro práci budete potřebovat Microsoft Visual Studio a měli byste mít alespoň základní znalosti XSLT a XAML. V menší míře budete potřebovat také základy C#.

PŘIDÁVÁNÍ A EDITACE VLASTNÍCH POLOŽEK

Jakmile máme instalaci hotovou, měli bychom mít celý projekt se všemi soubory otevřený a přístupný ve Visual Studiu, kde budeme dále pracovat. Projekt obsahuje velké množství souborů, ale pro naše účely jsou důležité *Config.daml*, obsah složky *Pages* a *Resources.resx* ve složce *Properties*.

Config.daml kromě jiného kontroluje, jaké stránky se mají zobrazovat v editoru, které z nich jsou povinné nebo nepovinné, využití exportérů, importérů atd. Představte si jej jako menu pro nastavení editoru.

¹ <https://github.com/Esri/arcgis-pro-metadata-toolkit>

² <https://github.com/Esri/arcgis-pro-metadata-toolkit/wiki/Build-your-first-ArcGIS-Pro-metadata-editor-add-in>

Složka *Pages* pak obsahuje zdrojový kód pro konkrétní stránky editoru. Soubory, které zde najdete, vám umožňují kontrolovat, jak budou jednotlivé stránky vypadat a jak se budou chovat.

Resources.resx zase kontroluje textové popisy tlačítek, polí a jiných prvků stránky.

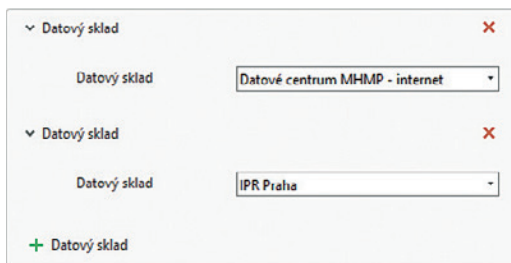
Vzhledem k velmi výrazné hloubce problému zde bohužel nemůžeme popisovat krok po kroku jednotlivá nastavení položek *Config.daml* nebo jiných souborů. Doporučujeme si je proto podrobně projít v předešlých odkazech na návody k toolkitu, abyste se dozvěděli, jak přidávat, upravovat nebo odebírat jednotlivé prvky editoru. Samotný návod obsahuje také několik ukázek přidávání vlastních položek jednoduchého typu a pro lepší pochopení problematiky doporučujeme projít si alespoň jednu z nich.

Na následujících řádcích se budeme věnovat přidání složitějšího metadatového prvku do editoru a zvládnutí příkladů z ukázek nám tak může v mnohém pomoci pochopit celý problém.

PŘIDÁNÍ VÍCENÁSOBNÉHO VYSKAKOVACÍHO MENU

Řekněme, že do editoru potřebujeme přidat vlastní metadatovou položku, která musí splňovat následující parametry:

- ▶ Je to vícenásobná položka, tzn. že může být použita 0 až n-krát.
- ▶ Položka je kontrolovaná, tzn. že může nabývat jen předem definované hodnoty.



Obr. 2. Příklad kontrolované a vícenásobné položky „Datový sklad“. Datový sklad může nabývat jen hodnoty obsažené v rozbalovací nabídce. Pro zadání libovolného počtu dalších hodnot z nabídky je nutné položku přes ovladač plus „+“ zduplikovat, v případě potřeby křížkem „x“ zrušit.

Položku potřebujeme přidat do existující stránky editoru (návod pro vytvoření zcela nové stránky najdete v návodu v kapitole *Create a custom page*). V případě IPR jsme potřebovali takovou položku přidat do stránky *Overview - Topics*

& *Keywords (Přehled - Kategorie & Klíčová slova)*. Musíme tak lokalizovat příslušný soubor XAML ve složce *Pages* v našem projektu ve Visual Studiu. Jelikož názvy souboru XAML nejsou identické jako názvy stránek editoru, je nutné najít takový, který alespoň přibližně odpovídá. V našem případě to je *keywords.xaml*. Kontrolou obsahu souboru pak dokážeme identifikovat, jestli jsme zvolili správně.

UserControl.Resources a Grid

Soubor má dvě hlavní části, *UserControl.Resources* a *Grid*. *UserControl.Resources* slouží pro definici xml data providerů, tedy xml elementů, které se do metadat mají zapsat. *Grid* pak kontroluje, co a jak se zobrazí na stránce. Když si obě části projedeme, shledáme, že *grid* obsahuje dané elementy editoru a *UserControl.Resources* obsahuje k nim náležící xml elementy. Prvky *gridu* musí odkazovat na xml element v *UserControl.Resources*. Bez této provazby by se námi zvolené hodnoty nazapsaly do metadat.

Prvním krokem, který proto musíme udělat, je vytvořit si xml element v *UserControl.Resources*, do kterého se zvolené hodnoty budou zapisovat. Na IPR jsme si takto vytvořili xml element s názvem *DatastoreInfo*, který obsahuje subelement *DatastoreDescription*. Další důležitou součástí je také *x:Key*, který slouží jako identifikátor xml prvku a slouží k spárování s prvkem v *gridu*. V našem případě je to *XmlRecord_DatastoreDesc* a výsledný xml element tak vypadá jako na obrázku 3.

V sekci *UserControl.Resources* máme hotovo a teď se přesuneme do *gridu*. Zvolíme si, do které části *gridu* chceme, aby byl náš nový prvek umístěn (tak jak prvky v *gridu* následují po sobě v kódu, tak jsou poskládány v editoru), a máme dvě možnosti, jak takový prvek vytvořit. Buďto si jej kompletně napíšeme sami, anebo můžeme zkopírovat už existující prvek a jenom si ho upravit dle potřeb. Jelikož metadatový editor obsahuje všechny možné variace metadatových elementů, tak je jednodušší identifikovat, který z nich chceme, kde se ve složce *Pages* nachází, zkopírovat ho a vložit do stejné stránky, v jaké jsme si vytvořili xml element a ve které pracujeme. Obdobný prvek, jaký chceme přidat, najdete například na stránce *Metadata - Maintenance (Metadata - Údržba)* a jmenuje se *Update Scope*. Když si proto identifikujeme daný prvek ve souboru XAML ve složce *Pages*, tak ho stačí zkopírovat a vložit

```
<!-- datastore -->
<XmlDataProvider x:Key="XmlRecord_DatastoreDesc" XPath="/" IsAsynchronous="False" IsInitialLoadEnabled="True">
  <x:XData>
    <ANY xmlns="">
      <DatastoreInfo editorExpand="True">
        <DatastoreDescription/>
      </DatastoreInfo>
    </ANY>
  </x:XData>
</XmlDataProvider>
```

Obr. 3. Vytvoření příslušného xml elementu.

```

<!-- datastore -->
<Border Style="{DynamicResource EditorBorder2Style}">
  <StackPanel>
    <ListBox Style="{DynamicResource EditorNoScrollListBoxStyle}" ItemsSource="{Binding XPath=DatastoreInfo[DatastoreDescription]}">
      <ListBox.ItemsPanel>
        <ItemsPanelTemplate>
          <StackPanel/>
        </ItemsPanelTemplate>
      </ListBox.ItemsPanel>
      <ListBox.ItemTemplate>
        <DataTemplate>
          <Expander Style="{DynamicResource EditorExpander2Style}"
AutomationProperties.AutomationId="{Binding RelativeSource={RelativeSource FindAncestor, AncestorType={x:Type ListBoxItem}}, Converter={StaticResource
_lbiToAutoIdConverter}, ConverterParameter=TopicsAndKeywordsPage_ThemeKeywordGroup}">
            <Expander.Header>
              <DockPanel HorizontalAlignment="{Binding HorizontalAlignment,
RelativeSource=[RelativeSource FindAncestor,
AncestorType={x:Type ContentPresenter}],
Mode=OneWayToSource}" LastChildFill="True">
                <Button DockPanel.Dock="Right" Click="DeleteItem" Style="{DynamicResource DeleteButton}"
Name="TopicsAndKeywordsPage_ThemeKeywordDelete">
                  <Image Style="{DynamicResource DeleteImageStyle}" Source="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/
GenericDeleteRed16.png"/>
                </Button>
                <Label Style="{DynamicResource EditorHExpanderLabel}" Content="{x:Static r:Resources.DATA_STORE}"/>
              </DockPanel>
            </Expander.Header>
            <ListBox Style="{DynamicResource EditorNoScrollListBoxStyle}" ItemsSource="{Binding XPath=DatastoreDescription}">
              <ListBox.ItemTemplate>
                <DataTemplate>
                  <StackPanel Style="{DynamicResource ExpanderBodyStyle}">
                    <tp:Custom_Datastore/>
                  </StackPanel>
                </DataTemplate>
              </ListBox.ItemTemplate>
            </ListBox>
          </Expander>
        </DataTemplate>
      </ListBox.ItemTemplate>
    </ListBox>

    <Button Tag="DatastoreDesc" Click="AddRecordByTagToLocal" Style="{DynamicResource AddButton}"
Name="ResourceDatastore_DatastoreAdd">
      <StackPanel VerticalAlignment="Center" Orientation="Horizontal">
        <Image Style="{DynamicResource AddImageStyle}" Source="pack://application:,,,/ArcGIS.Desktop.Resources;component/Images/GenericAddGreen16.png"/>
        <TextBlock Style="{DynamicResource ButtonTextStyle}" Text="{x:Static r:Resources.DATA_STORE}"/>
      </StackPanel>
    </Button>
  </StackPanel>
</Border>

```

Obr. 4. Kód pro nový element datového skladu. Části zvýrazněné zelenou barvou je třeba upravit. Výsledkem je nový prvek v editoru na obrázku č. 2.

do naší stránky. Následně je ještě potřeba upravit některé části tohoto elementu.

Úprava kódu elementu

Úpravy kódu můžeme rozdělit na tři části. Nejprve musíme navázat náš kód na xml element vytvořený dříve. Propojení dosáhneme úpravou prvního, třetího a pátého zvýrazněného prvku z obrázku 4 (zelená barva). První a třetí prvek na obrázku mají definovaný parametr *binding path* pro *listbox*. Tento parametr rozhoduje o tom, kam se budou metadata zapisovat v rámci xml elementu. Pátý prvek dělá v podstatě to stejné, jenom pro tlačítko přidání nového záznamu.

Kromě toho musíme také upravit popisky, a to změnou druhého a šestého (posledního) zvýrazněného prvku v obrázku. Popisky kontroluje prvek *Resources.Data_Store* z obrázku je pak námi vytvořený název popisku v souboru *Resources.resx*, o kterém už byla řeč. Jak správně zavést

nový popisek do *Resources.resx* a následně ho pak použít se můžete dozvědět také v návodu v kapitole *Create a custom page*.

Posledním krokem je definice samotného obsahu vyskakovací nabídky. Jelikož se jedná o složitější prvek, který může být použit vícenásobně, nebo vůbec, tak jej nemůžeme definovat přímo v *Keywords.xaml*, jako se to dá udělat u jednodušších prvků, ale musíme vytvořit nové soubory XAML obsahující asociované soubory C# kontrolující chování prvků. Tvorba těchto souborů je poněkud náročná, vy ale můžete využít odkazu na zdrojový kód našeho projektu³ a jen si zkopírovat předmětné XAML i s CS podsoubory (*Custom_Datastore.xaml* a *Custom_DatastoreBase.xaml*), mírně si je poupravit, přejmenovat a dostanete funkční prvek metadat. Podrobnosti fungování těchto souborů bohužel není v rozsahu článku možné pokrýt, avšak pomocníkem, který vám může ulehčit práci, je sledování všech výskytů klíčového slova *Datastore* a jeho nahrazení vlastní

³ <https://github.com/vyvoj-iprpraha/ArcGISpro-metadata-editor>

```

-<DatastoreInfo>
  -<DatastoreDescription>
    <DatastoreType value="IPR Praha" />
  </DatastoreDescription>
</DatastoreInfo>
-<DatastoreInfo>
  -<DatastoreDescription>
    <DatastoreType value="Datové centrum MHMP - internet" />
  </DatastoreDescription>
</DatastoreInfo>
-<DatastoreInfo>
  -<DatastoreDescription>
    <DatastoreType value="Datové centrum MHMP - mepnet" />
  </DatastoreDescription>
</DatastoreInfo>

```

Obr. 5. Zobrazení zápisu datového centra v metadatech ArcGIS Pro.

hodnotou ve všech XAML a souvisejících podsouborech. Zkrátka je nutné nastudovat si logiku daných souborů, jejich provazby a fungování, a pak zjistíte, co je nutné upravit, aby vám prvek fungoval.

Kontrolovat, zdali se všechno přepisuje a funguje, jak má, můžete přímo v ArcGIS Pro při zobrazení metadat. Abyste změny mohli vidět, je v *Config.daml* nutné změnit nastavení pro xml viewer na *XML.XSL*.

VALIDACE OPROTI INSPIRE – VYUŽITÍ SOUBORŮ .XSLT

Kromě přidání nových elementů/prvků do metadatového editoru ArcGIS Pro bylo pro IPR také nezbytné, aby byla metadata zaslána na validaci oproti české verzi INSPIRE. Pro export metadat do XML, které se validuje, využívá ArcGIS Pro soubory XSLT sloužící k transformaci z jednoho XML zápisu do jiného. Museli jsme si proto vytvořit vlastní soubor XSLT (*ArcGIS_2ISO19139_CustomExporter.xslt*), na který jsme pak v *Config.daml* odkázali. Podrobný návod je opět možné najít v dokumentaci Esri, ve článku *Customize the metadata translators*⁴.

Kromě toho můžete v našem projektu na GitHubu³ také pozorovat všechny změny zachycené v souboru *ArcGIS_2ISO19139_CustomExporter.xslt* a případně je

použít či zkopírovat pro vaše účely. Jak projekt na GitHubu, tak upravované soubory obsahují další komentáře upřesňující některé části kódu.

ZÁVĚREM

Posledním a nejjednodušším krokem celého procesu, který vás čeká, je vybudování celého projektu a následné přidání samotného add-inu do ArcGIS Pro. Pokud vaše organizace přechází se správou metadat do ArcGIS Pro a zároveň máte vlastní metadatové položky, které v žádném z defaultních metadatových profilů nenajdete, tak je úprava editoru vhodným řešením vaší situace. Rozsah a možnosti toolkitu jsou samozřejmě mnohem mnohem větší a i samotné úpravy vzpomínaných souborů mohou jít do mnohem větší hloubky, než je zde nastíněno. Můžete například kontrolovat dokonce i popisky tooltipů pomocí *Definitions.resx* nebo upravovat jazykové mutace vašich prvků.

Ve článku jsme sice nemohli jít do přílišných detailů, avšak pokud budete potřebovat přidat vlastní prvky do metadatového editoru, všechno je velmi dobře popsáno v návodu na GitHubu⁵, případně se můžete inspirovat také naším repozitářem³, kde najdete podrobnější vysvětlení a další typy vlastních prvků, které jsme přidávali. ◀◀

Ing. Róbert Spál, Institut plánování a rozvoje hl. m. Prahy
Kontakt: spal@ipr.praha.eu

Projekt na GitHubu Esri: <https://github.com/Esri/arcgis-pro-metadata-toolkit>

Projekt na GitHubu IPR: <https://github.com/vyvoj-iprpraha/ArcGISpro-metadata-editor>

⁴ <https://github.com/Esri/arcgis-pro-metadata-toolkit/wiki/Customize-the-metadata-translators>

⁵ <https://github.com/Esri/arcgis-pro-metadata-toolkit/wiki/Build-your-first-ArcGIS-Pro-metadata-editor-add-in>