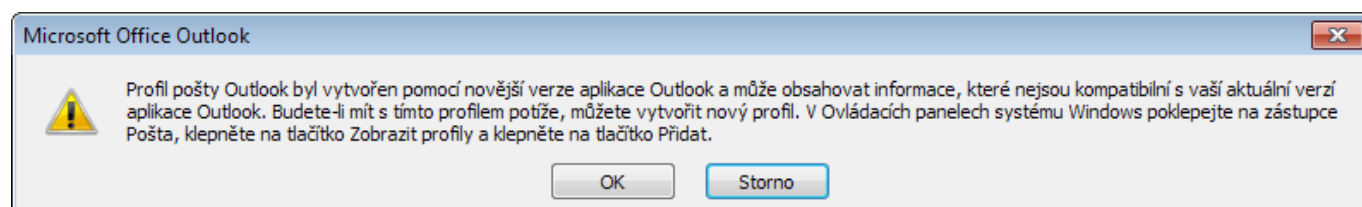


Tímto článkem si odskočím z Excelu do Outlooku a popíšu svůj gurmánský zážitek s jeho objektovým modelem.

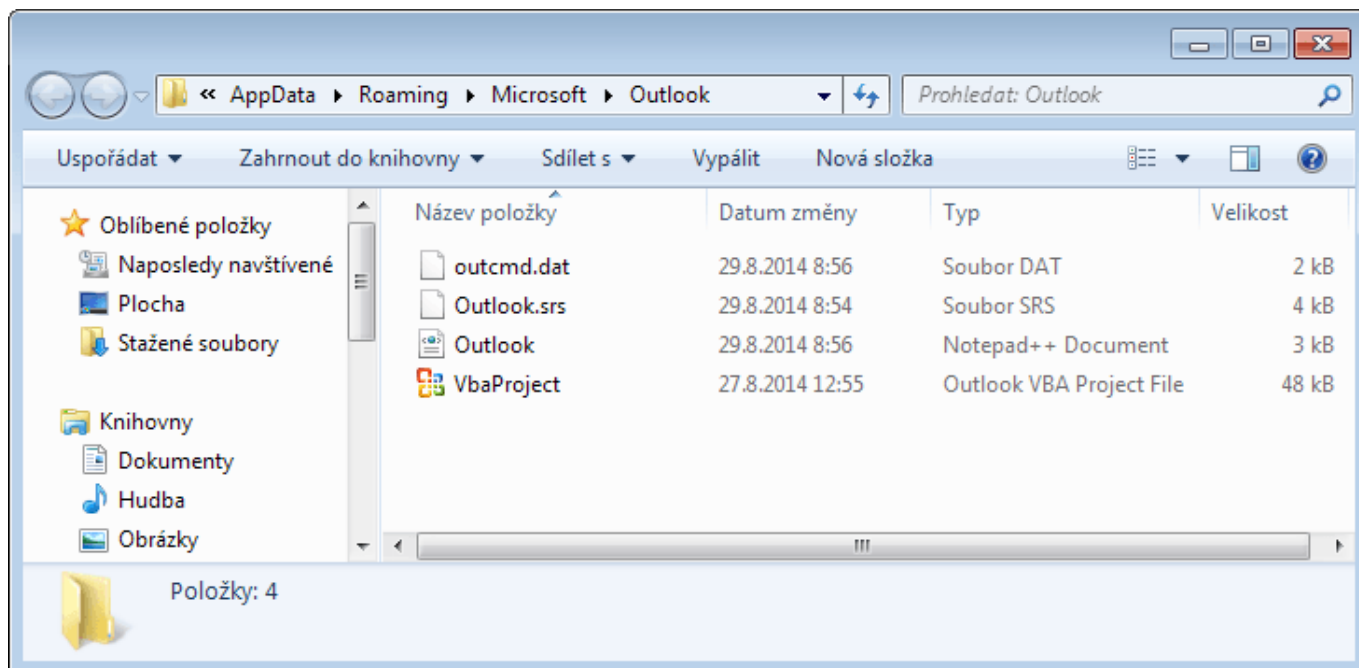
Před pár dny mi přišel jednoduchý požadavek klienta umístit do Microsoft Outlooku 2007 (a novější verze) tlačítko odkazující se na webovou stránku. No fajn, tak udělám doplněk a je to, říkal jsem si. Až na ten malý háček, že nějaké VBA doplňky Outlook vůbec nezajímají, orientuje se pouze na doplňky COM. Do instalace Visual Studia se mi vůbec nechtělo, ale co se dalo dělat. Jenže jsem u toho jaksi nepřemýšlel. K dispozici jsem měl verzi VS 2008, která nepodporovala momentální verzi Outlooku 2010. A tak jsem si zazálohoval poštu, sestřelil Outlook 2010 a nahradil ho verzí 2007, což samozřejmě dvakrát korektní není (pošťák se se soubory nové verze nesrovná nikdy).



Downgrade Microsoft Outlook 2010 na 2007

V tu chvíli jsem si uvědomil dávno zapomenutou věc, že Outlook 2007 ještě neobsahoval Pás karet a bude nutné zpracovat minimálně dvě řešení. Navíc při kompilaci projektu pro Outlook 2007 na mě VS zařvalo chybou. Pár minut jsem ještě zkoušel přeinstalovat PIA a pak jsem COM variantu doplňku shodil ze stolu.

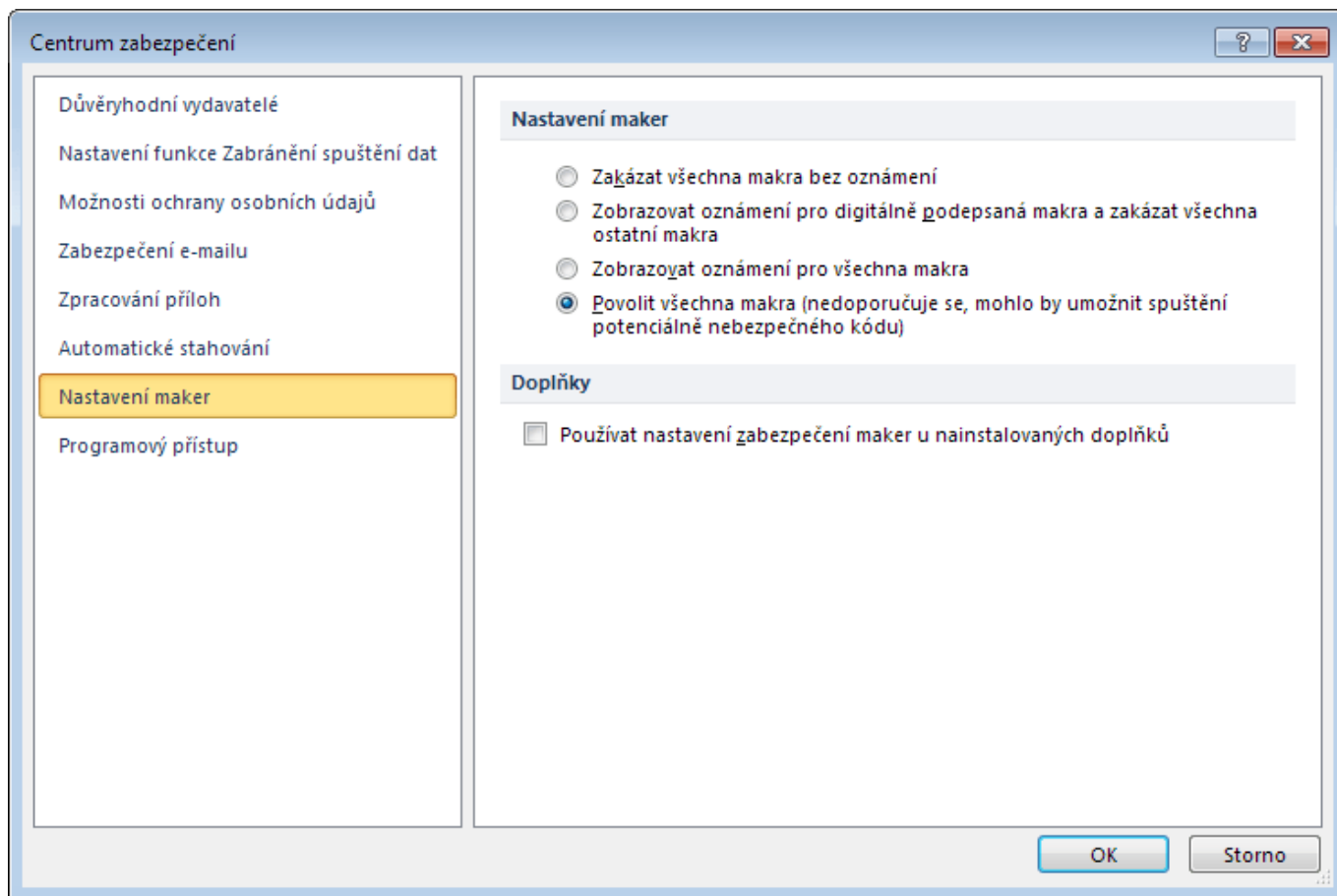
Zbývalo jediné – starý dobrý CommandBar s tlačítkem, a to i za cenu toho, že se bude na ribbonu zobrazovat pod kartou Doplňky. Bylo mi také jasné, že se klient nevyhne zásahu do zabezpečení maker v nastavení Outlooku. Docela mi trvalo, než jsem přišel na to, že k automatickému spouštění makra po startu Outlooku slouží procedura `Application_Startup` umístěná v modulu (třídě) `ThisOutlookSession`. Stejně tak jsem musel dohledat, že `CommandBars` se odvíjí od objektu `Outlook.ActiveExplorer` a že tlačítko není provázáno na makro přes `OnAction`, ale má svou událost `Click`. Hned vzápětí mi docvaklo, že Outlook a jeho VBA neobsahuje `FollowHyperlink` a přechod na webovou stránku ve výchozím prohlížeči bude muset obstarat API a funkce `ShellExecute`. Zapudil jsem v tu chvíli myšlenku, že bych měl rozlišovat 32 a 64bitovou variantu deklarace této funkce. Kód VBA se poslušně uložil do datové složky Outlooku pod názvem `VbaProject.OTM` a já se rozhodl jej poslat klientovi k vyzkoušení.



Datová složka pro Microsoft Outlook

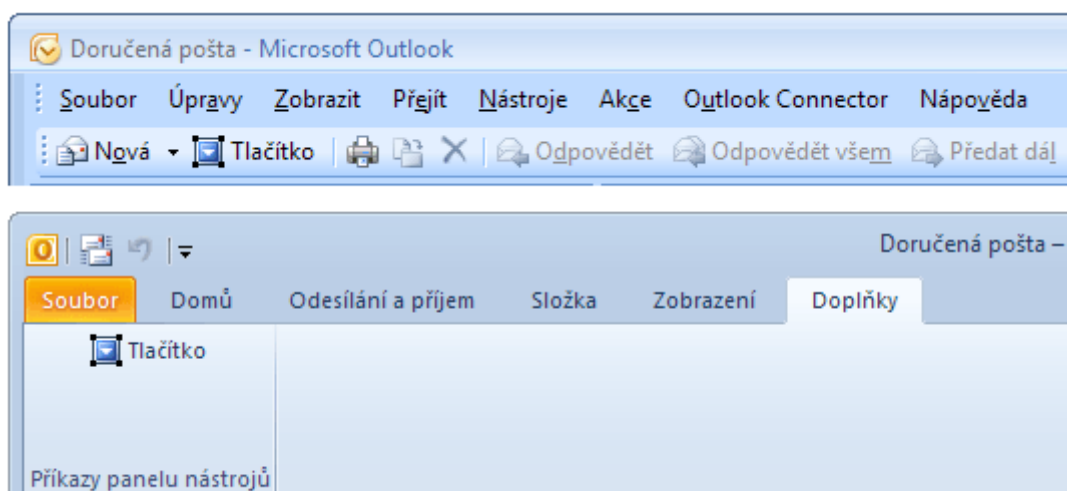
I když jsem mu popsal umístění souboru, složku nenašel a vyžadoval instalátor. Spáchal jsem tedy malý soubor VBScriptu, jenž využil systémovou proměnnou `%APPDATA%` a překopíroval soubor, kam patřil.

V další reakci jsem se dočkal tvrzení, že se tlačítko nezobrazuje. Chvilka testování a musel jsem mu dát za pravdu. A diskuse na internetu také. Motor `Application_Startup` prostě napoprvé nezažehne, jak má. Jako by nebyla tato událost správně zaregistrována. Desítky uživatelů se dlouhé roky trápí tím, co je příčinou. Povolují veškerá nastavení zabezpečení, podepisují makra, diskutují o „event handleru“ a dělají psí kusy. A Microsoft to má doslova na háku.



Zabezpečení v Microsoft Outlook 2010

Jedno z řešení nabízelo vytvořit zástupce Outlooku s přepínačem /autorun a makro si tak vynutit. Přepínač byl ale s příchodem SP2 pro Outlook 2003 zakázán. Další z nich - /altvba - umožňuje definovat alternativní OTM soubor s makry ke spuštění, ovšem s tím, že pro své fungování vyžaduje existenci klíče v registru (můžete být klidní, ani tohle motor nenahodí). Toho času jsem měl všeho už opravdu plné zuby. Dozvěděl jsem se, že postačí ručně překompilovat VBA projekt. Bohužel to pořád neřešilo prvotní spuštění bez zásahu uživatele. Jednoduše jsem doplnil kód o veřejné makro, které uživatel napoprvé spustí z dialogu Makra přes ALT+F8 (nemohl jsem ho nutit do karty Vývojář) a přidal další bod instrukce (1. zabezpečení maker, 2. instalace souboru do datové složky Outlooku a 3. první ruční spuštění). S kyselým obličejem tedy toto povídání zakončím zahřměním: Co je mi do nějakého blběho červeného tlačítka a Nového Zélandu...



Tlačítko v Microsoft Outlook 2007 a 2010