Multilingual App Toolkit

Standards and multilingual software development
29, April 2015

Jan Anders Nelson
Microsoft Corporation
Overview

- Visual Studio integration
- Common localization workflow
- Industry standard XLIFF support
- Integrates with Microsoft Translator
- Pseudo translation
- 3rd party translation provider support
- Translation Editor
Multilingual App Toolkit Workflow

Enable ‘MAT’ on project
Add translation languages
Generate Translations
Run & Test
Demo
Resources
Multilingual App Toolkit Resources

http://aka.ms/matinstallv4
Download v4.0

http://aka.ms/matblog
Announcements

http://aka.ms/matvoice
User voice site

multilingual@Microsoft.com
Email for the MAT team.
Thank you!
Globalization - Best practices

Use Windows.Globalization namespace

- Windows.Globalization.NumberFormatting
  - DecimalFormatter
  - PercentFormatter
  - CurrencyFormatter
  - PermilleFormatter

- Windows.Globalization.DateTimeFormatting
  - DateTimeFormatter

- Windows.Globalization
  - Calendar
Globalization - Avoid

• Handcrafted formatting
  • String.Format("{0}/{1}/{2}", Month, Day, Year)
  • String.Format("${0:0.00}", 100.00);

• Ignoring user’s language preferences
  • PrimaryLanguageOverride is permanent – use with care.

• Creating specialized calendar UI controls
Localization – Best practices

- Build-in localization into your app
  - It always harder to retro-fix globalization

- Use resource files for all localizable strings
  - However, hard-code strings that should not be localized

- Translations may require 40% more space

- Utilize resource binding
  - Store: x:Uid for XAML, data-win-res for HTML

- ResourceLoader.GetString("MyResourceId")

- Use resource qualifiers for culture specific images
  - Avoid embedded text in images
MAT – Best practices

• Develop using Pseudo to find issues early
  • Pseudo is not shown by default, add it by searching for language “qps-PLOC”

• Designate a primary world-ready project contact
  • Ideally this is a development lead with globalization and localization knowledge

• Use a subset of languages during development
  • Ideally this will included Arabic, Asian, and Latin based languages

• Not every developer needs to run MAT all the time
  • A single person can manage and check-in initial language for the team

• Include professional translation in overall workflow