Babylon – Feature list

Localizing Visual Studio projects and ASP.NET Core projects

- Localize Windows Forms projects
- Localize WPF projects
- Localize ASP.NET projects
- Localize Silverlight projects (limited to resource files)
- Localize Windows Phone Apps
- Localize Windows RT Apps
- Localize .NET Standard projects
- Localize .NET Core projects
- Localize ASP.NET Core projects
- Multi-framework project support

Localizing Angular projects

- Localize Angular projects using the standard Angular localization support (i18n).
- Localize Angular projects using the ngx-translate resource provider.

Localizing Generic projects

- Plugin-based ResourceProviders to read and write resource files from any source and location
- ResourceProviders for XLIFF 1.2, XML, JSON, ngx-translate and Java .properties resource files

Localization in the release cycle

- Create new translation projects by scanning the source project
- Add/remove locales to and from project
- Separated translation status and quality status for every resource string in every language
- “Do not translate” status for resource strings to avoid unnecessary translation
- Editing of invariant language possible
- Support for custom cultures
- Automatic verification of translations: string format operators for several programming languages, inconsistent translations, punctuation and other
- String length plausibility check
- Writing of resource files in selected locales
- Automatic check-out and check-in of resource files from source control management systems
- Direct generation of unsigned or signed satellite assemblies in selected locales
- Direct generation of localized Angular Builds

- Synchronize existing localization project with the source project. All changed strings will be marked as changed, new strings will be added and obsolete strings removed
- Separate, free Translator Edition allows localization by external translators simply by sending them the localization project file
- Import of localized strings from another project to consolidate all translations back into the master localization project
- Automatically remove unused resource strings from invariant resource file to avoid unnecessary localization effort
- Quality review wizard to quickly review all not approved resource strings
- Swap locales. Allows changing the invariant language of a project in a couple of seconds
- Automatic Translation using Google Translator
- Automatic Translation using MS Azure
- Automatic Translation using DeepL
- Pseudo Translation
- Automatic extraction of string literals from source code and generation of resources
- Create, edit and manage translation memories and use them to automatically translate new strings
- Export/Import Translation Memory to and from TMX files
- Export/Import to and from XLIFF files
- Batch copy string of one locale to another

Command line interface

- Open a solution
- Synchronize Solution with Visual Studio Projects
- Synchronize Solution with Angular Projects
- Synchronize Solution with Generic projects
- Write translated Resx files
- Generate Satellite Assemblies
- Perform Angular build
- Auto translate using Google, MS Azure, DeepL or Pseudo Translator
- Generate Translation Packages for external translation
- Import Translation Packages after translation

Editing

- Multi level Undo/Redo
- Preview dialog to preview translated dialogs
- Find
- Inline Spell Checker
- Set font size in main translation view for easier reading
- Highlight translation status using different background colors

- Highlight quality status using different background colors

User Interface

- Modern ribbon based user interface. Simple and intuitive.
- Highly customizable main grid view with sorting, grouping and filtering of resource strings
- Graphical representation of translation and quality status situation