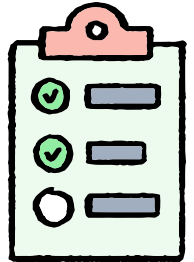


Technical localization checklist



Pre-development (planning & design)

- ☐ Define localization goals per market (languages, regions, compliance requirements)
- ☐ Identify which content, screens, and features need localization
- ☐ Choose supported locales and fallback logic (e.g., en-US → en)
- ☐ Include localization in product specs and design mockups
- ☐ Plan for right-to-left (RTL) support, even if not immediately needed
- ☐ Allocate space in UI designs for text expansion (30–50% longer)
- ☐ Avoid embedding text in images or video without a localization plan
- ☐ Document non-textual locale variations (e.g., date/time, addresses, currency)

Development (internationalization)

- ☐ Externalize all user-facing text into separate resource files
- ☐ Use string keys with consistent naming conventions ("**button.save**, not **btn_1**")
- ☐ Support Unicode/UTF-8 encoding for all content and inputs
- ☐ Implement locale-aware formatting for dates and times, numbers and currencies, measurement units (metric/imperial)
- ☐ Handle pluralization rules using ICU or localization libraries
- ☐ Build UI layouts that adjust dynamically to different text lengths
- ☐ Implement RTL layout logic and test for mirroring support
- ☐ Support dynamic locale switching if required by UX
- ☐ Use language tags (e.g., **lang="fr-CA"**) where appropriate

Localization process setup

- ☐ Choose and integrate a translation management system (TMS)
- ☐ Set up automated string sync between codebase and TMS
- ☐ Provide translators with: screenshots for context, in-line comments or notes, and character limits (if needed)
- ☐ Define review and QA process (LQA, functional testing)
- ☐ Implement pseudo-localization for early testing
- ☐ Build workflow to track translation status across locales

QA and testing

- ☐ Run in-context QA for all localized versions
- ☐ Verify layout integrity (no text overflow, truncation, or clipping)
- ☐ Check formatting and logic for dates, currencies, and numbers
- ☐ Ensure all strings are translated
- ☐ Validate RTL behavior (mirrored layout, correct text alignment)
- ☐ Use native speakers for final content review when possible
- ☐ Test dynamic components (forms, dropdowns, messages) in all locales
- ☐ Ensure fallback language works if locale-specific strings are missing

Maintenance and scaling

- ☐ Set up version control for all string files
- ☐ Remove unused strings to reduce translator overload
- ☐ Keep localization as part of the development/release cycle
- ☐ Track string changes across releases and notify translators
- ☐ Monitor for locale-specific bugs post-launch
- ☐ Maintain documentation for developers and translators
- ☐ Regularly audit your translation management system and workflows for improvements