

Power Pages: Mehrsprachige Websites per Plugin für GitHub Copilot CLI und Claude Code

Executive Summary

Kennst du den Satz „Französisch machen wir in Phase 2“? Das ist der Satz, mit dem in Lenkungsausschüssen Mehrsprachigkeit beerdigt wird. Phase 2 kommt nie, das Budget ist weg, und der Kunde in Lausanne liest dein Portal mit dem Browser-Übersetzer, der aus „Antrag stellen“ zuverlässig etwas Peinliches macht. Genau an dieser Stelle setzt Microsoft jetzt an.

Das **Power-Pages-Plugin für GitHub Copilot CLI und Claude Code** hat Mitte Oktober 2026 den neuen Skill `/add-localization` bekommen. Der Agent macht damit aus einer bestehenden Power-Pages-Code-Site eine mehrsprachige Website: Er erkennt das Framework (React, Vue oder Angular), legt dir einen Plan zur Freigabe vor, installiert das passende Paket, zieht alle sichtbaren Texte in Sprachdateien, baut eine Sprachauswahl ein und testet jede Sprache im Browser. Unterstützt werden **mehr als 43 Sprachen**, einschließlich rechts-nach-links-Schriften wie Arabisch und Hebräisch.

Die Kehrseite: Der Skill übersetzt **nur die Oberfläche**. Dataverse-Inhalte, Umgebungssprachen und die Systemmeldungen von Power Pages bleiben deine Baustelle. Und das Ganze läuft als Preview. Für DACH-Portale mit Deutsch, Französisch, Italienisch und Englisch sinkt der Aufwand trotzdem spürbar – vorausgesetzt, du behandelst die KI-Übersetzung als Rohentwurf und nicht als Endprodukt.

Fakten: Das Wichtigste in 30 Sekunden

- Neuer Skill `/add-localization` im Power-Pages-Plugin (Marketplace `microsoft/power-platform-skills`).
- Mehr als 43 Sprachen, inklusive RTL-Layout für Arabisch und Hebräisch.
- Frameworks: React, Vue, Angular (Runtime-Modus). Astro ist vorerst nicht unterstützt.
- Übersetzungen wahlweise KI-generiert oder als leere Dateien für dein Übersetzungsbüro.
- Auch direkt in `/create-site` integriert: Sprachen gleich beim Anlegen der Site auswählen.

Worum geht es im Detail?

Ein bisschen Hintergrund, damit du die Tragweite einordnen kannst. Power Pages – früher Power Apps Portals, davor Dynamics 365 Portals, davor Adxstudio – war lange eine Welt aus Liquid-Templates, Webseiten-Datensätzen und Content Snippets. Mehrsprachigkeit hieß dort: Sprachpakete in Dataverse aktivieren, jede Seite pro Sprache pflegen und beten, dass niemand ein Snippet vergisst. Funktioniert, ist aber ungefähr so sexy wie eine Steuererklärung in dreifacher Ausfertigung.

Seit einiger Zeit gibt es zusätzlich **Code Sites**: Single-Page-Applications in React, Vue, Angular oder Astro, die du lokal entwickelst und über die Power Platform CLI nach Power Pages hochlädst.

Dataverse, Tabellenberechtigungen und Web-Rollen bleiben, das Frontend ist aber ganz normaler Webcode. Und normaler Webcode ist genau das, was Coding-Agenten gut können.

Im Februar 2026 hat Microsoft deshalb das Power-Pages-Plugin für GitHub Copilot CLI und Claude Code als Public Preview veröffentlicht. Installiert wird es in beiden Werkzeugen mit `/plugin marketplace add microsoft/power-platform-skills` und `/plugin install power-pages@power-platform-skills`. Voraussetzung sind Node.js ab Version 18, eine aktuelle PAC CLI, die Azure CLI, angemeldete Sitzungen und eine Umgebung mit Power Pages. Das Plugin kennt die Power-Pages-Eigenheiten – Tabellenberechtigungen, Web-Rollen, Site-Einstellungen, Web-API-Muster – und fragt vor jeder kritischen Änderung nach.

Skill	Was er tut
<code>/create-site</code>	Legt ein Projekt in React, Vue, Angular oder Astro an und öffnet eine Live-Vorschau
<code>/setup-datamodel</code>	Schlägt ein Dataverse-Schema vor und legt es nach Freigabe an
<code>/integrate-webapi</code>	Ersetzt Mock-Daten durch echten Web-API-Code
<code>/setup-auth, /create-webroles</code>	Anmeldung über Microsoft Entra ID und rollenbasierte Zugriffe
<code>/add-seo</code>	Sitemaps und Meta-Tags
<code>/deploy-site, /activate-site</code>	Hochladen nach Power Pages und öffentliche URL bereitstellen
<code>/add-localization</code> (neu)	Mehrsprachigkeit inklusive Sprachauswahl, RTL und Browser-Tests

Wie der Skill vorgeht

Spannend ist weniger, dass der Agent übersetzt, sondern wie kontrolliert er es tut. Wer schon einmal einem Agenten beim fröhlichen Umschreiben von 140 Dateien zugesehen hat, weiß das zu schätzen. Der Skill läuft in acht Phasen, und zwei davon sind harte Schranken.

So arbeitet /add-localization

Acht Phasen, zwei harte Freigabe-Gates – und erst danach sieht irgendein Besucher eine neue Sprache

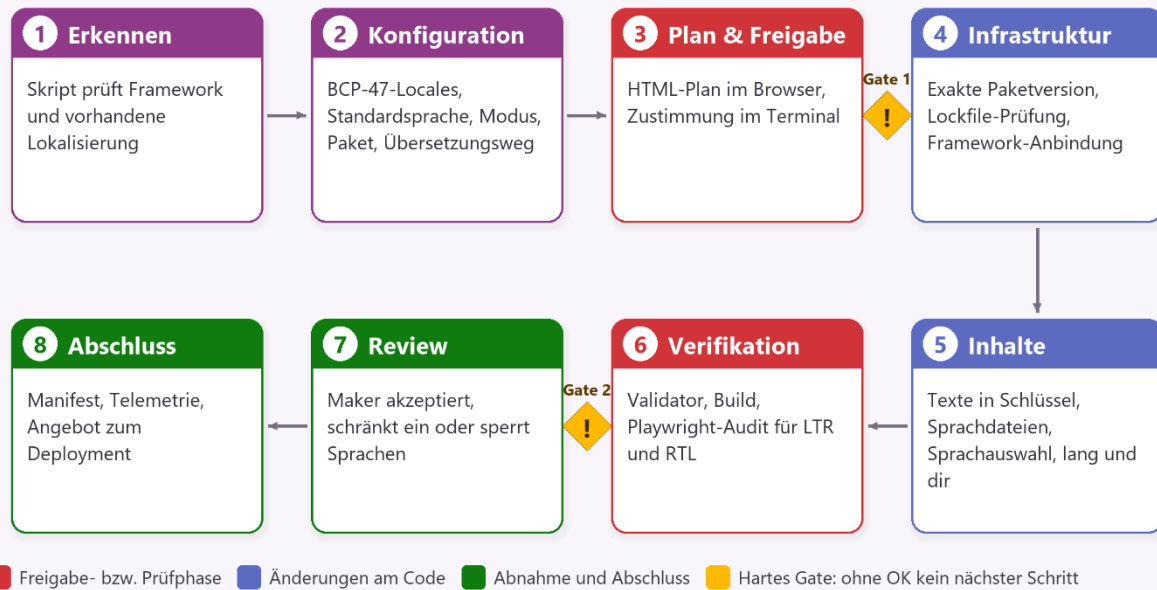


Abbildung 1: Der Ablauf von /add-localization mit Freigabe- und Prüf-Gate

Zuerst prüft ein Skript, welches Framework im Projekt steckt und ob es schon Lokalisierung gibt. Sind die Hinweise widersprüchlich, musst du entscheiden – der Agent rät nicht. Danach fragt er in fester Reihenfolge ab: Sprachen als **BCP-47-Tags** (also **de-DE**, **de-CH**, **fr-CH**, **it-CH**), die Standardsprache, den Modus, das Paket und den Übersetzungsweg. Das Ergebnis landet als HTML-Plan im Ordner **docs/** und öffnet sich im Browser. Erst wenn du im Terminal zustimmst, wird irgendetwas installiert.

Bei den Paketen ist Microsoft erfreulich paranoid: Ein Skript prüft Kompatibilität, Wartungszustand, Lizenz und Modus. Harte Fehler lassen sich nicht übersteuern, bei Lizenzfragen musst du ausdrücklich bestätigen. Installiert wird eine exakt validierte Version, deren Herkunft über die **package-lock.json** abgesichert wird. Für Angular empfiehlt der Skill [@jsverse/transloco](#). Alle Einstellungen hält er im Manifest **.powerpages-localization.json** fest.

Dann folgt die eigentliche Arbeit: sichtbare Texte in sprechende Schlüssel extrahieren, Sprachdateien befüllen, Sprachauswahl einbauen, **lang**- und **dir**-Attribute setzen. Vorhandene Übersetzungen bleiben unangetastet. Anschließend prüfen ein unabhängiger Validator, der Build und ein Playwright-Audit die gerenderte Seite in beiden Schreibrichtungen. Eine Sprache, die durchfällt, taucht weder in der Auswahl noch in der automatischen Erkennung noch im Produktiv-Build auf. Und bei KI-Übersetzungen ist ein Warnhinweis Pflicht: Die Texte können Fehler enthalten und müssen vor der Veröffentlichung geprüft werden.

Was genau übersetzt wird – und was nicht

Was der Skill übersetzt – und was dein Problem bleibt

Links: dieselbe Seite in Deutsch und Arabisch. Rechts: die Grenze zwischen Agent und Projektteam.

contoso.powerappsportals.com

LOGO

DE v

Willkommen im Portal

dir="ltr"

LTR

contoso.powerappsportals.com

AR v

LOGO

dir="rtl"

RTL

Erledigt der Skill

- ✓ Sichtbare UI-Texte in Schlüssel extrahieren
- ✓ Sprachdateien: KI-Übersetzung oder leer
- ✓ Sprachauswahl in die Navigation einbauen
- ✓ lang- und dir-Attribute, RTL-Layout
- ✓ Build und Browser-Test je Sprache

Bleibt beim Projektteam

- ✗ Dataverse-Datensätze (Produkte, Artikel, FAQ)
- ✗ Sprachen der Umgebung aktivieren
- ✗ Power-Pages-Systemmeldungen
- ✗ Juristische Freigabe: Impressum, Datenschutz
- ✗ Muttersprachliches Review vor Go-live

43+ Sprachen · React, Vue, Angular (Runtime) · Astro: noch nicht · Status: Preview

Abbildung 2: Gleiche Seite, zwei Schreibrichtungen – und die Grenze der Zuständigkeit

Thema	Status	Was das für dich heißt
React, Vue	Unterstützt	Runtime-Lokalisierung mit zentralem Locale-Koordinator
Angular	Teilweise	Nur Runtime; der statische Modus ist vorübergehend deaktiviert
Astro	Nicht unterstützt	Der Skill bricht vor der Planung ab
npm mit package-lock.json	Pflicht	Yarn- oder pnpm-Projekte ohne npm-Lockfile fallen raus
Dataverse-Datensätze	Nicht enthalten	Produktkataloge, Artikel, Stammdaten übersetzt du selbst
Systemmeldungen, Umgebungssprachen	Nicht enthalten	Klassische Power-Pages-Konfiguration bleibt nötig

4

Wichtig: UI-Lokalisierung ist nicht Content-Lokalisierung

Dein Portal besteht typischerweise zu 20 Prozent aus Buttons und Menüs und zu 80 Prozent aus Daten. Der Skill kümmert sich um die 20 Prozent. Wer den Rest vergisst, hat am Ende ein französisches Menü über deutschen Produktbeschreibungen. Sieht international aus, ist es aber nicht.

Was sind Chancen? Was sind Risiken?

Die größte Chance ist banal: Mehrsprachigkeit wandert von „Phase 2“ in den Sprint. Was früher ein bis zwei Wochen Handarbeit war – Strings suchen, Schlüssel erfinden, i18n-Bibliothek anbinden, Sprachumschalter basteln, RTL-Fehler jagen –, erledigt der Agent in einer Sitzung. Bei einem Kundenportal eines Schweizer Verbands haben wir früher allein für das saubere Herausziehen der Texte aus einer gewachsenen React-App vier Personentage gebraucht. Und danach noch einen Tag, weil jemand „Speichern“ an 37 Stellen hart codiert hatte, in drei unterschiedlichen Schreibweisen.

Die zweite Chance ist Disziplin. Plan, Freigabe, Validator, Browser-Test: Der Skill zwingt dir einen Prozess auf, den viele Teams nie hatten. Das ist ein bisschen wie ein Fitnesstrainer, der dich nur ins Studio lässt, wenn du vorher den Gesundheitscheck machst. Nervig, aber du überlebst länger.

Chancen	Risiken
Mehrsprachigkeit kostet Stunden statt Wochen	KI-Übersetzung klingt flüssig – und ist trotzdem falsch
RTL-Märkte werden ohne Spezialwissen erreichbar	RTL-Feinschliff bleibt manuelle Prüfarbeit
Einheitliche Schlüssel und Struktur über alle Sites	Nur UI – Dataverse-Inhalte bleiben einsprachig
Erzwungene Freigaben und Tests als Qualitätsnetz	Preview-Status: Verhalten kann sich ändern
Leere Sprachdateien als saubere Übergabe an Übersetzer	Agent arbeitet mit deinen CLI-Anmeldedaten

Kommen wir zum unterhaltsamen Teil. **KI-Übersetzungen** sind gefährlich, weil sie so überzeugend aussehen. Ein Fehler, der holpert, fällt auf. Ein Fehler, der elegant formuliert ist, geht live. In einem Portal für Fördermittelanträge hat eine automatische Übersetzung einmal aus „Frist gewahrt“ ein französisches „Frist verpasst“ gemacht. Niemand hat es bemerkt, bis die ersten Antragsteller ihre Unterlagen gar nicht mehr eingereicht haben. Der Agent hatte seine Arbeit gemacht, das Team seine nicht.

Warnung: Juristische Texte gehören nicht in den Autopiloten

Impressum, Datenschutzhinweise, Nutzungsbedingungen und Einwilligungstexte lässt du bitte nicht vom Agenten übersetzen und dann ungeprüft veröffentlichen. Wenn der Datenschutzbeauftragte die französische Fassung zum ersten Mal in einer Abmahnung liest, ist der gesparte Personentag sehr teuer geworden. Für solche Texte: leere Sprachdateien wählen und an Profis geben.

Beim Thema **RTL** gilt: Der Playwright-Audit findet viel, aber nicht alles. Gespiegelte Pfeile, Icons mit Leserichtung, Formulare mit Telefonnummern, Datumsfelder, eingebettete Drittkomponenten – der Skill dokumentiert seine Befunde, die visuelle Abnahme bleibt aber ausdrücklich Handarbeit. Nichts sagt so überzeugend „Wir nehmen den Nahen Osten ernst“ wie ein Checkout, dessen Weiter-Button mutig in die falsche Richtung zeigt und den Kunden zurück in den Warenkorb schickt. Für immer.

Warnung: Der Agent hat deine Schlüssel

Das Plugin arbeitet mit deinen angemeldeten PAC- und Azure-CLI-Sitzungen. Wenn du als globaler Administrator gegen die Produktivumgebung angemeldet bist und „mach mal“ tippst, hast du nicht KI-Entwicklung eingeführt, sondern russisches Roulette mit Autovervollständigung. Entwicklungsumgebung, eigenes Konto, minimale Rechte.

Und dann die Preview. Microsoft hat schon in der Skill-Dokumentation Pfade als „vorübergehend nicht verfügbar“ markiert, etwa statisches Angular oder Astro. Das ist ehrlich, heißt aber auch: Was heute funktioniert, kann sich mit dem nächsten Plugin-Update ändern. Für Kundenprojekte mit festem Abnahmetermin ist das kein Ausschlusskriterium, aber ein Grund, Plugin-Version und Lockfile einzufrieren.

Was müssen wir jetzt schon vorbereiten?

Die gute Nachricht: Das meiste, was du vorbereiten musst, ist nicht technisch. Die schlechte: Genau deshalb wird es gern vergessen. Hier die Liste, die wir in Projekten gerade abarbeiten.

Aufgabe	Warum	Wer
Sprachstrategie festlegen	Welche Locales, welche Standardsprache, de-CH ohne ß oder nicht?	Fachbereich, Marketing
Übersetzungsweg pro Inhaltstyp	UI per KI mit Review, Rechtstexte per Fachübersetzer	Projektleitung, Recht
Projekt auf npm mit Lockfile umstellen	Ohne package-lock.json verweigert der Skill die Arbeit	Entwicklung
Dataverse-Inhalte planen	Mehrsprachige Spalten oder Tabellen modellieren	Architektur
Glossar anlegen	Damit „Antrag“ nicht in jeder Sprache dreimal anders heißt	Fachbereich
Governance für Coding-Agenten	Lizenzen, Datenabfluss, Konten, Freigaberegeln	IT-Sicherheit, Betriebsrat
Muttersprachliches Review	Pro Sprache eine Person mit Abnahmerecht	Fachbereich

Besonders unterschätzt wird die **Governance**. Claude Code und GitHub Copilot CLI sind Werkzeuge, die auf dem Rechner von Entwicklerinnen und Entwicklern laufen, Code lesen, Befehle ausführen und mit Cloud-Modellen sprechen. Das ist für viele Datenschutzbeauftragte ungefähr so

beruhigend wie ein Praktikant mit Generalschlüssel. Kläre vorab, welches Werkzeug mit welcher Lizenz erlaubt ist, welche Repositories der Agent sehen darf und dass er nie gegen Produktion läuft. Wenn ihr einen Betriebsrat habt: Der möchte auch gefragt werden, bevor ein Agent Logs über die Arbeitsweise schreibt.

Tipp: Klein anfangen, mit Pilot-Site

Nimm eine bestehende, nicht kritische Code-Site, lass `/add-localization` mit zwei Sprachen laufen und wähle bewusst einmal KI-Übersetzung und einmal leere Dateien. Dann vergleichst du Aufwand und Qualität mit echten Zahlen statt mit Bauchgefühl. Den HTML-Plan aus `docs/` kannst du direkt in die Projektdokumentation übernehmen.

Wichtig: Schweizer Kundschaft? Dann denk an de-CH

Für Deutsch in der Schweiz ist `de-CH` die richtige Wahl, nicht `de-DE`. Dort schreibt man „Strasse“ statt „Straße“, und ein Portal, das Schweizer Kunden mit bundesdeutschem ß begrüßt, wirkt ungefähr so lokal wie ein Fondue aus der Mikrowelle. Leg die Regel ins Glossar, damit Agent und Review sie kennen.

Technisch solltest du außerdem die Werkzeugkette aktuell halten: Plugin aktualisieren, PAC CLI aktualisieren, Node.js ab Version 18. Wer bereits eine Site mit `/create-site` startet, kann die Sprachen dort gleich mitbestellen. Bestehende Sites bekommen sie über `/add-localization` nachgerüstet.

Häufig gestellte Fragen

Funktioniert `/add-localization` auch mit klassischen Power-Pages-Sites ohne Code-Site?

Nein. Der Skill arbeitet mit Power-Pages-Code-Sites, also Single-Page-Applications in React, Vue oder Angular, die über die Power Platform CLI bereitgestellt werden. Klassische Sites auf Basis von Liquid-Templates lokalisierst du weiterhin über Dataverse-Sprachen und mehrsprachige Inhaltsseiten.

Übersetzt der Skill auch Inhalte aus Dataverse?

Nein, er übersetzt ausschließlich die Texte der Benutzeroberfläche. Dataverse-Datensätze, Umgebungssprachen und Power-Pages-Systemmeldungen musst du separat konfigurieren und übersetzen.

Wie zuverlässig sind die KI-Übersetzungen des Power-Pages-Plugins?

Sie sind ein brauchbarer Rohentwurf, aber kein freigabefähiges Endprodukt. Microsoft schreibt selbst einen Warnhinweis vor, dass die Übersetzungen Fehler enthalten können und vor der Veröffentlichung geprüft werden müssen; für Rechtstexte solltest du leere Sprachdateien wählen und professionell übersetzen lassen.

Brauche ich für das Plugin eine Lizenz für GitHub Copilot oder Claude Code?

Ja, du benötigst Zugang zu GitHub Copilot CLI oder Claude Code mit dem jeweils passenden Abonnement. Dazu kommen Node.js ab Version 18, die Power Platform CLI, die Azure CLI und eine Power-Platform-Umgebung mit Power Pages.

Werden rechts-nach-links-Sprachen wie Arabisch vollständig automatisch umgesetzt?

Der Skill setzt die Schreibrichtung, passt das Layout an und prüft die gerenderte Seite automatisiert im Browser. Die visuelle Abnahme von Icons, Formularen und Drittkomponenten bleibt aber ausdrücklich ein manueller Schritt.