

# WPF:agent — WPForms Source Wrap Proposal

## Overview

WPF:agent proposes evolving WPForms into a new class of agent-operable forms.

Instead of treating forms purely as static data collectors, this model allows forms to act as guided intake and fulfillment systems while preserving the familiar WPForms user interface.

The goal is not to replace forms, but to enhance forms as fulfillment systems.

Instead of treating forms only as data collection tools, they become:

- guided intake systems
- structured intent interpreters
- media-aware submission builders
- fulfillment launch points

## Core Idea

Traditional form workflow:

User fills fields  
→ form validates  
→ entry submitted

Proposed WPF:agent workflow:

User expresses intent  
→ agent assists completion  
→ form structure adapts  
→ supporting media gathered  
→ submission becomes a fulfillment packet

This creates a new interaction model:

**message <=> media**

The user's message shapes the form interaction, and the form structure helps refine the message until a valid submission is produced.

## Why WPForms Is the Ideal Platform

WPForms already provides the strongest foundation for this evolution.

It already includes:

- conditional logic
- multi-page flows
- file uploads
- partial saves
- strong UX
- a large install base
- extensible hooks

This means WPForms already acts as a structured intake surface.

WPF:agent proposes exposing that structure to an assistance layer.

## What a Source Wrap Layer Does

The Source Wrap layer would expose limited, structured information about a form to an agent system.

Possible exposed information:

- form schema
- field definitions
- page structure
- conditional logic
- validation state
- partial entry state
- uploaded media

This allows an agent to guide the user without bypassing WPForms' existing flow.

The form remains the primary interface.

The agent simply helps operate it.

## Example Agent Capabilities

An agent connected to the Source Wrap could:

- explain confusing fields
- suggest how to answer questions
- convert narrative text into structured fields
- detect missing required information

- recommend supporting documents
- organize uploaded media
- validate logical consistency
- prepare the final submission packet

The form remains intact, but the interaction becomes significantly easier.

## Media-Aware Forms

Most forms allow attachments, but those attachments are passive.

WPF:agent treats media as active submission evidence.

Examples:

- screenshots
- documents
- photos
- generated reports

These can be analyzed and connected to specific form fields.

The final submission becomes a structured evidence bundle, not just a list of answers.

## Fulfillment-Oriented Submissions

Instead of ending at “entry saved”, the form can produce:

- verification workflows
- automated task creation
- case records
- administrative review packets
- downstream integrations

This aligns with many real-world use cases where a form submission initiates a complex process.

## Potential Use Cases

WPF:agent would be especially powerful in domains such as:

### Compliance intake

Structured regulatory information and documentation.

## **Client onboarding**

Complex onboarding workflows requiring explanation and documentation.

## **Tax preparation**

Guided intake for complicated tax forms.

## **Claims and evidence collection**

Evidence-heavy claim submissions.

## **Grant applications**

Narrative plus structured submission workflows.

## **Consent systems**

Regulated consent flows requiring verification.

# **Implementation Approach**

This proposal does not require fundamental changes to WPForms.

A minimal implementation could include:

- 1. Form schema exposure**  
Structured description of fields and pages.
- 2. Entry state access**  
Partial entry data and validation status.
- 3. Event hooks**  
Form progression events and field completion events.
- 4. Media linking**  
Associate uploads with form fields.
- 5. Agent interaction endpoints**  
Safe XHR interface for assistance.

This could exist as:

- a WPForms addon
- a developer extension
- an experimental feature

## Why This Matters

Forms are one of the most universal interfaces on the internet.

However, many important forms are:

- confusing
- incomplete
- abandoned
- filled incorrectly

Agent-assisted forms can dramatically improve:

- completion rates
- accuracy
- accessibility
- user confidence

## Relationship to Existing WPForms Philosophy

This concept aligns strongly with WPForms' existing goals:

- making forms easier
- improving user experience
- reducing friction

WPF:agent extends those principles into the agent era.

## Proposal Summary

WPF:agent proposes that WPForms evolve from a **form builder** into an **operable intake platform** where forms become intelligent, media-aware, and fulfillment-oriented.

## Closing Thought

Forms are one of the most important interfaces on the web.

With the addition of intelligent assistance, they can evolve from static questionnaires into interactive fulfillment systems.

WPForms is uniquely positioned to lead that transformation.