

V2PipelineConfig

September 4, 2026

V2PipelineConfig

Kind: Interface

Source: [atloria-monorepo/packages/doc-automation/src/v2/orchestrator.ts](#)

Properties

Property	Type
repository	{

```
url: string;
branch: string;
username?: string;
password?: string;
paths: {
  frontend?: string;
  backend?: string;
};
/** Parser configuration for parser-core integration */
parsers?: {
  /** Explicit parser names to use (e.g. ['odoo-xml', 'python', 'django']). If omitted, auto-de
  explicit?: string[];
  /** Additional file patterns to include beyond parser defaults */
  additionalPatterns?: string[];
  /** Use parser-core for parsing (default: true). Set false for legacy regex fallback. */
  useParserCore?: boolean;
};
```

```
} | | project | {
id: string;
name: string;
description?: string;
} | | ai | {
/** AI provider to use (default: 'claude') /
provider?: 'openai' | 'claude' | 'azure-responses';
```

```

// Claude/Anthropic config
azureEndpoint?: string;
azureApiKey?: string;
anthropicApiKey?: string;
// OpenAI config
openaiApiKey?: string;
azureOpenaiEndpoint?: string;
azureOpenaiApiKey?: string;
azureOpenaiDeployment?: string;
// Azure Responses API config (gpt-5.6-terra by default, overridable via
AZURE_OPENAI_MODEL)
azureResponsesEndpoint?: string;
azureResponsesApiKey?: string;
azureResponsesApiVersion?: string;
// Per-stage model override (used with azure-responses provider)
stageModels?: {
analysis?: string;
synthesis?: string;
structure?: string;
planning?: string;
generation?: string;
verification?: string;
screenshots?: string;
};
// Shared config
model?: string;
maxTokens?: number;
} | / documentation / {
targetAudience: 'beginner' | 'intermediate' | 'advanced';
tone: 'formal' | 'friendly' | 'technical';
language: string;
maxDocuments: number;
// User-suggested categories from project config (AI can add/remove based on
codebase)
suggestedCategories?: string[];
} | / output / {
workDir: string;
outputDir: string;

```

```

} | / options / {
skipScreenshots?: boolean;
skipUpload?: boolean;
skipVerification?: boolean;
resumeFromCheckpoint?: boolean;
skipQualityGate?: boolean;
/* Run screenshots in background (non-blocking). Default: true /
asyncScreenshots?: boolean;
/* Skip AI verification for docs that passed quality gate. Default: true /
skipVerificationForPassingDocs?: boolean;
/* Screenshots only — skip doc generation, read existing docs from DB /
screenshotsOnly?: boolean;
/* Use V4d structure generation — replaces Stages 4+5+6 with parser-based
approach.
* Uses slim synthesis (1 AI call) + 3 parallel structure sessions.
* ~3 min, ~$0.50 vs current ~37 min, ~$8. /
useV4dStructure?: boolean;
/* Enable agent-powered verification — uses AI agent with code tools to verify
* generated docs against source code. More thorough than Stage 11's LLM-only
check.
* Runs after Quality Gate, before Stage 11. Default: false (opt-in). /
enableAgentVerification?: boolean;
/* Use agent-writer for Stage 10 — AI agent explores source code AND writes docs
* in the same session. More accurate than GeneratorService because the agent reads
* actual code while writing. Requires Azure Responses API credentials.
* Default: false (opt-in). /
useAgentWriter?: boolean;
} | / businessDomain / 'ERP' | 'SaaS' | 'E-commerce' | 'CMS' | 'Generic' /
/ businessContext / BusinessContext / / contextSources / { type: 'pdf' | 'url' | 'file' |
'text'; path: string; content?: string }[] / / rateLimiter / {
/* Max concurrent AI requests. Default: 5 for Azure, 30 for others /
maxConcurrent?: number;
/* Min delay between requests in ms. Default: 1000 for Azure, 25 for others /
minDelayMs?: number;
} | / screenshots / {
/* URL of the screenshot-worker microservice /
workerUrl?: string;
/* Site URL for screenshots (defaults to siteUrl from config) /

```

```

siteUrl?: string;
/* Login URL for authentication /
loginUrl?: string;
/* Login credentials /
credentials?: {
  username: string;
  password: string;
  usernameSelector?: string;
  passwordSelector?: string;
  submitSelector?: string;
  successSelector?: string;
};
/* Viewport configuration /
viewport?: {
  width: number;
  height: number;
};
/* Screenshot format /
format?: 'png' | 'jpeg' | 'webp';
/* Screenshot quality (for jpeg/webp) /
quality?: number;
/* Maximum concurrent capture sessions /
maxConcurrency?: number;
/* Azure blob storage container name /
azureContainerName?: string;
/* Whether to check for duplicate screenshots /
checkDuplicates?: boolean;
/* Frontend framework hint (e.g. 'angular', 'react', 'vue') — used for URL validation /
framework?: string;
} | | onScreenshotsComplete | (stats: { captured: number; failed: number; replaced:
number; totalPlaceholders: number }) => void | | onScreenshotProgress | (progress:
{ captured: number; failed: number; total: number; currentUrl?: string }) => void |
| upload | {
/* Prisma client for database access /
prisma: any; // Using 'any' to avoid circular dependency with @prisma/client
/* Organization ID for document ownership /
organizationId: string;
/* User ID who created the documents /

```

```
createdById: string;
/* Job ID for tracking which job created documents /
jobId?: string;
/* Target audience slug (e.g., 'user', 'developer') /
audienceSlug?: string;
/* Optional function to upload images to Azure via assets API /
uploadImage?: (buffer: Buffer, filename: string, mimeType: string) => Promise;
/* Batch configuration /
batch?: {
  maxRetries?: number;
  retryDelay?: number;
  maxConcurrency?: number;
  requestsPerSecond?: number;
  uploadScreenshotsFirst?: boolean;
};
/* When true, documents with groundingScore < 0.3 are excluded from upload.
Default: false /
skipUnverifiableContent?: boolean;
/* DocVersion ID (Phase 1) — when set, upload operates in create-only mode */
docVersionId?: string;
} |
```