Deployment Guide

> **Source File:** generated-documents\implementation-guides\deployment-guide.md
> **Generated:** 30/07/2025 at 06:58:07
> **Generated by:** Requirements Gathering Agent - PDF Converter

Deployment Guide

Generated by adpa-enterprise-framework-automation v3.2.0
Category: implementation-guides
Generated: 2025-07-30T01:09:15.566Z
Description: Application deployment guide and procedures

Deployment Guide

Project: === PROJECT README ===

ADPA - Advanced Document Processing & Automation Framework

Monorepo CI failing
vercel api error

npm package 3.2.0
node >=18.0.0
TypeScript 5.7.2
License MIT
API-First TypeSpec

> ***Previously known as Requirements Gathering Agent (RGA)***

ADPA is a modular, standards-compliant enterprise automation framework for AI-powered document generation, project management, and business analysis. Built with TypeScript and Node.js, it provides both CLI and REST API interfaces for generating professional documentation following industry standards including BABOK v3, PMBOK 7th Edition, and DMBOK 2.0.

🚀 Key Features

Enterprise Standards Compliance

- 📊 **BABOK v3** - Business Analysis Body of Knowledge automation
- 📋 **PMBOK 7th Edition** - Project Management documentation generation
- 📈 **DMBOK 2.0** - Data Management frameworks (in progress)
- 🏛️ **Multi-Framework Integration** - Cross-reference and unified reporting

AI-Powered Generation

- 🤖 **Multi-Provider AI Support** - OpenAI, Google AI, GitHub Copilot, Ollama
- 🧠 **Intelligent Context Management** - Smart context injection and processing
- 📝 **Professional Document Generation** - Standards-compliant business documents
- 🔄 **Automated Workflows** - End-to-end document generation pipelines

Enterprise Integration

- 🌐 **Production-Ready REST API** - TypeSpec-generated OpenAPI specifications
- 📚 **Confluence Integration** - Direct publishing to Atlassian Confluence
- 📊 **SharePoint Integration** - Microsoft SharePoint document management
- � **Adobe Document Services** - Professional PDF generation and document intelligence
- �🔧 **CLI & Web Interface** - Multiple interaction modes

Compliance & Security

- 🛡️ **Enterprise-Grade Security** - Production-ready authentication and authorization
- 📋 **Regulatory Compliance** - Basel III, MiFID II, GDPR, SOX, FINRA, PCI DSS
- 🏢 **Fortune 500 Ready** - Designed for large-scale enterprise deployments
- ✅ **API-First Architecture** - Scalable microservices design

📦 Installation

NPM Package (Recommended)

```
npm install -g adpa-enterprise-framework-automation
```

From Source

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run build
```

Docker (Coming Soon)

```
docker pull adpa/enterprise-framework:latest
```

🎯 Quick Start

1. CLI Usage

```
# Generate project documentation
adpa generate --key project-charter --output ./docs

# Start the API server
adpa-api

# Initialize Confluence integration
adpa confluence init

# Initialize SharePoint integration
adpa sharepoint init
```

2. API Server

```
# Start the Express.js API server
npm run api:start

# Access API documentation
open http://localhost:3000/api-docs
```

3. Admin Web Interface

```
# Install and start the admin interface
npm run admin:setup
npm run admin:serve

# Access at http://localhost:3001
```

🛠️ Configuration

Environment Setup

1. **Copy the environment template:**

```
cp .env .env.local  # Create your local configuration
```

2. **Configure your preferred AI provider:**

Option 1: Google AI Studio (Recommended - Free Tier)

```
# Set the active provider
CURRENT_PROVIDER=google-ai

# Get your API key from: https://makersuite.google.com/app/apikey
GOOGLE_AI_API_KEY=your-google-ai-api-key-here
GOOGLE_AI_MODEL=gemini-1.5-flash
```

Option 2: GitHub AI (Free for GitHub Users)

```
# Set the active provider
CURRENT_PROVIDER=github-ai

# Get your token from: https://github.com/settings/tokens
GITHUB_TOKEN=your-github-personal-access-token
GITHUB_ENDPOINT=https://models.github.ai/inference/
REQUIREMENTS_AGENT_MODEL=gpt-4o-mini
```

Option 3: Azure OpenAI (Enterprise)

```
# Set the active provider
CURRENT_PROVIDER=azure-openai-key

# Configure Azure OpenAI
AZURE_OPENAI_ENDPOINT=https://your-resource.openai.azure.com/
AZURE_OPENAI_API_KEY=your-azure-openai-api-key
AZURE_OPENAI_DEPLOYMENT_NAME=your-deployment-name
AZURE_OPENAI_API_VERSION=2024-02-15-preview
```

Option 4: Ollama (Local)

```
# Set the active provider
CURRENT_PROVIDER=ollama

# Configure Ollama (requires local installation)
OLLAMA_ENDPOINT=http://localhost:11434/api
OLLAMA_MODEL=deepseek-coder:latest
```

AI Provider Configuration

ADPA supports multiple AI providers with automatic failover:

- **Google AI Studio** - Free tier with generous limits (1M-2M tokens)
- **GitHub AI** - Free for GitHub users with gpt-4o-mini access
- **Azure OpenAI** - Enterprise-grade with Entra ID authentication
- **Ollama** - Local models for privacy-focused deployments

Provider Priority: The system will automatically fallback to available providers if the primary provider fails.

📚 Framework Support

BABOK v3 (Business Analysis)

✅ **Production Ready**

- Requirements Elicitation & Analysis
- Stakeholder Analysis & Management
- Business Analysis Planning
- Requirements Life Cycle Management
- Strategy Analysis
- Requirements Analysis & Design Definition
- **Solution Evaluation**: Evaluate implemented solutions for business value, performance, and alignment with stakeholder needs. Supports continuous improvement and benefit realization tracking.
- **Underlying Competencies**: Describes the foundational skills, behaviors, and knowledge areas required for effective business analysis, as defined by BABOK v3.
- **Perspectives**: Outlines the various perspectives (Agile, BI, IT, Business Architecture, BPM) and how to tailor business analysis practices for each context, as defined by BABOK v3.
- Enterprise Analysis
- **Introduction Business Analysis Body of Knowledge**: Provides an overview, checklist, and summary of all BABOK documents, including coverage gaps and improvement suggestions. This document is generated as the starting point for BABOK-based documentation in ADPA.

PMBOK 7th Edition (Project Management)

✅ **Implemented**
<<<<<<< Updated upstream

- Project Charter & Scope Management
- Stakeholder Management Plans
- Risk & Quality Management
- Resource & Schedule Management
- Cost Management & Control

DMBOK 2.0 (Data Management)

✅ **Production Ready**

- Data Governance Frameworks (see `data-governance-framework` document type)
- Data Stewardship & Roles (see `data-stewardship-roles-responsibilities` document type)
- Data Modeling Standards (see `data-modeling-standards` document type)
- Data Quality Management (see `data-quality-management-plan` document type)
- Data Architecture & Quality
- Data Architecture & Modeling (see `data-architecture-modeling-guide` document type)

- Business Intelligence & Analytics Strategy (see `business-intelligence-strategy` document type)
- Master Data Management (see `master-data-management-strategy` document type)
- Metadata Management (see `metadata-management-framework` document type)
- Data Security & Privacy (see `data-security-privacy-plan` document type)
- Reference Data Management (see `reference-data-management-plan` document type)
- Data Storage & Operations (see `data-storage-operations-handbook` document type)
- Data Lifecycle Management (see `data-lifecycle-management` document type)

🏗️ Architecture

Core Components

```
ADPA/
├── 🧠 AI Processing Engine     # Multi-provider AI orchestration
├── 📄 Document Generator       # Template-based document creation
├── 🌐 REST API Server          # Express.js with TypeSpec specs
├── 💻 CLI Interface            # Yargs-based command line tools
├── 🔌 Integration Layer        # Adobe, Confluence, SharePoint, VCS
├── 🎛️ Admin Interface          # Next.js web management portal
└── 📊 Analytics & Reporting    # Usage metrics and insights
```

Technology Stack

- **Backend**: Node.js 18+, TypeScript 5.7+, Express.js
- **AI Integration**: OpenAI, Google AI, GitHub Copilot, Ollama
- **API**: TypeSpec, OpenAPI 3.0, Swagger UI

Generate Data Modeling Standards Guide (DMBOK)

adpa generate --key data-modeling-standards --format markdown

- **Frontend**: Next.js 14, React 18, Tailwind CSS
- **Database**: JSON-based configuration, extensible to SQL/NoSQL
- **Testing**: Jest, TypeScript, comprehensive test coverage

📖 Usage Examples

Document Generation

```
# Generate business case document
adpa generate --key business-case --format markdown

# Generate complete project charter
adpa generate --category project-charter --output ./project-docs
```

```
# Generate stakeholder analysis
adpa generate --key stakeholder-analysis --format json



# Generate Solution Evaluation (BABOK)
adpa generate --key solution-evaluation --format markdown

# Generate Underlying Competencies (BABOK)
adpa generate --key underlying-competencies --format markdown

# Generate Perspectives (BABOK)
adpa generate --key perspectives --format markdown

# Generate Introduction Business Analysis Body of Knowledge (BABOK)
adpa generate --key introduction-business-analysis-body-of-knowledge --format markdown

# Generate Data Governance Framework (DMBOK)
adpa generate --key data-governance-framework --format markdown

# Generate Data Stewardship and Roles & Responsibilities (DMBOK)
adpa generate --key data-stewardship-roles-responsibilities --format markdown

# Generate Data Quality Management Plan (DMBOK)
adpa generate --key data-quality-management-plan --format markdown

# Generate Master Data Management Strategy (DMBOK)
adpa generate --key master-data-management-strategy --format markdown

# Generate Data Architecture & Modeling Guide (DMBOK)
adpa generate --key data-architecture-modeling-guide --format markdown

# Generate Metadata Management Framework (DMBOK)
adpa generate --key metadata-management-framework --format markdown

# Generate Data Security & Privacy Plan (DMBOK)
adpa generate --key data-security-privacy-plan --format markdown

# Generate Reference Data Management Plan (DMBOK)
adpa generate --key reference-data-management-plan --format markdown

# Generate Data Storage & Operations Handbook (DMBOK)
adpa generate --key data-storage-operations-handbook --format markdown

# Generate Data Lifecycle Management Policy (DMBOK)
adpa generate --key data-lifecycle-management --format markdown

# Generate Document & Content Management Framework (DMBOK)
adpa generate --key document-content-management --format markdown

# Generate Business Intelligence & Analytics Strategy (DMBOK)
adpa generate --key business-intelligence-strategy --format markdown
```

API Usage

```
// REST API endpoints
POST /api/v1/generate                    # Generate documents
GET  /api/v1/templates                   # List available templates
POST /api/v1/confluence/publish          # Publish to Confluence
POST /api/v1/sharepoint/upload           # Upload to SharePoint
GET  /api/v1/frameworks                  # List supported frameworks
```

Integration Examples

```
# Adobe Document Services integration
npm run adobe:setup                      # Configure Adobe credentials
npm run adobe:demo-generation           # Run document generation demo
npm run adobe:example-basic             # Basic PDF generation example

# Confluence integration
adpa confluence oauth2 login
adpa confluence publish --document ./docs/project-charter.md

# SharePoint integration
adpa sharepoint oauth2 login
adpa sharepoint upload --folder "Project Documents" --file ./docs/

# Version control integration
adpa vcs commit --message "Generated project documentation"
adpa vcs push --remote origin
```

Portfolio/Program Stakeholder Analysis

Generate a stakeholder analysis at the portfolio or program level (multi-project, business unit, or enterprise-wide):

```
adpa generate --key portfolio-stakeholder-analysis --format markdown
```

This document provides a comprehensive analysis of stakeholders across multiple projects, programs, or business units, supporting portfolio management best practices.

🧪 Testing

```
# Run all tests
npm test

# Test specific providers
npm run test:azure
npm run test:github
npm run test:ollama

# Performance testing
npm run test:performance
```

```
# Integration testing
npm run test:integration
```

🏢 Enterprise Features

Compliance Standards

- **Financial**: Basel III, MiFID II, FINRA, CFTC, FCA, BaFin
- **Security**: GDPR, SOX, PCI DSS, ISO 27001, ISO 9001
- **Industry**: Healthcare (HIPAA), Government (FedRAMP)

Enterprise Integration

- **Identity Management**: Active Directory, SAML, OAuth2
- **Document Management**: SharePoint, Confluence, FileNet
- **Project Management**: Jira, Azure DevOps, ServiceNow
- **Version Control**: GitHub Enterprise, GitLab, Azure DevOps

Scalability & Performance

- **Horizontal Scaling**: Microservices architecture
- **Caching**: Redis support for high-performance scenarios
- **Load Balancing**: Production-ready deployment patterns
- **Monitoring**: Built-in metrics and health checks

📁 Project Structure

```
requirements-gathering-agent/
├── 📁 src/                          # TypeScript source code
│   ├── 🎯 cli.ts                    # Main CLI entry point
│   ├── 🌐 server.ts                 # Express.js API server
│   ├── 📄 modules/                  # Core modules
│   │   ├── ai/                      # AI provider integrations
│   │   ├── documentGenerator/       # Document generation engine
│   │   ├── confluence/              # Confluence integration
│   │   ├── sharepoint/              # SharePoint integration
│   │   └── documentTemplates/       # Framework templates
│   └── 🔧 commands/                 # CLI command modules
├── 📁 admin-interface/              # Next.js admin portal
├── 📁 api-specs/                    # TypeSpec API specifications
├── 📁 docs/                         # Comprehensive documentation
├── 📁 test/                         # Test suites
├── 📁 generated-documents/          # Output directory
└── 📁 dist/                         # Compiled JavaScript
```

🤝 Contributing

We welcome contributions! Please see our Contributing Guide for details.

Development Setup

```
git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run dev          # Start development mode
npm run build        # Build for production
npm test            # Run tests
```

Code Standards

- **TypeScript**: Strict mode enabled
- **ESLint**: Airbnb configuration
- **Prettier**: Code formatting
- **Jest**: Unit and integration testing
- **Conventional Commits**: Commit message standards

📋 Roadmap

Q2 2025

- ✅ DMBOK 2.0 implementation
  🔄 Docker containerization
  🔄 Kubernetes deployment templates
  🔄 Advanced analytics dashboard

Q3 2025

- 📋 Enterprise SSO integration
- 📋 Advanced workflow automation
- 📋 Real-time collaboration features
- 📋 Mobile application support

📞 Support & Documentation

- 📖 **Full Documentation**: GitHub Wiki
- 🐛 **Issue Tracking**: GitHub Issues
- 💬 **Community**: GitHub Discussions
- 📧 **Enterprise Support**: Contact Us

📄 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- **Industry Standards**: PMI (PMBOK), IIBA (BABOK), DAMA (DMBOK)

- **AI Providers**: OpenAI, Google, GitHub, Ollama community
- **Enterprise Partners**: Fortune 500 beta testing organizations
- **Open Source Community**: Contributors and feedback providers

---

Built with ❤️ for Enterprise Automation

🌟 Star us on GitHub | 📦 npm Package | 📖 Documentation

=== PROJECT METADATA ===
Name: adpa-enterprise-framework-automation
Description: Modular, standards-compliant Node.js/TypeScript automation framework for enterprise requirements, project, and data management. Provides CLI and API for BABOK v3, PMBOK 7th Edition, and DMBOK 2.0 (in progress). Production-ready Express.js API with TypeSpec architecture. Designed for secure, scalable, and maintainable enterprise automation.
Version: 3.2.0
Dependencies: @adobe/pdfservices-node-sdk, @azure-rest/ai-inference, @azure/identity, @azure/msal-node, @azure/openai, @google/generative-ai, @microsoft/microsoft-graph-client, @types/mongoose, axios, bcryptjs, compression, cors, dotenv, express, express-rate-limit, express-validator, express-winston, form-data, glob, helmet, joi, jsonwebtoken, marked, mongoose, morgan, multer, node-fetch, openai, puppeteer, swagger-ui-express, ts-node, uuid, winston, yargs, zod
Dev Dependencies: @jest/globals, @redocly/cli, @types/bcryptjs, @types/compression, @types/cors, @types/express, @types/glob, @types/jest, @types/jsonwebtoken, @types/morgan, @types/multer, @types/node, @types/node-fetch, @types/swagger-ui-express, @types/uuid, @typespec/compiler, @typespec/http, @typespec/json-schema, @typespec/openapi3, @typespec/rest, ajv, jest, rimraf, ts-jest, typescript, webpack-cli
Available Scripts: build, copy-configs, start, api:start, dev, clean, test, test:providers, test:performance, test:azure, test:github, test:ollama, test:failover, test:unit, prepublishOnly, admin:install, admin:dev, admin:build, admin:start, admin:setup, admin:serve, confluence:init, confluence:test, confluence:oauth2:login, confluence:oauth2:status, confluence:oauth2:debug, confluence:publish, confluence:status, sharepoint:init, sharepoint:test, sharepoint:oauth2:login, sharepoint:oauth2:status, sharepoint:oauth2:debug, sharepoint:publish, sharepoint:status, api:compile, api:watch, api:format, api:lint, api:docs, api:serve-docs, api:demo, api:server, babok:generate, pmbok:generate, dmbok:generate, framework:multi

=== SAMPLE-BUSINESS-REQUIREMENTS.MD (planning) ===
Path: ADPA\demo\sample-business-requirements.md
Relevance Score: 90

Sample Business Requirements Document

Project Overview

This is a sample document for demonstrating the ADPA hub features. This document contains typical business requirements content that will showcase how our new hub system works in Microsoft Word.

Timeline Requirements

The project should be completed in the following phases:

Phase 1: Planning (Week 1-2)

- Requirements gathering
- Stakeholder interviews
- Initial analysis

Phase 2: Development (Week 3-8)

- System design
- Core functionality implementation
- Testing framework setup

Phase 3: Testing (Week 9-10)

- Unit testing
- Integration testing
- User acceptance testing

Phase 4: Deployment (Week 11-12)

- Production deployment
- User training
- Go-live support

Business Requirements

Functional Requirements

1. **User Authentication**: System must support secure user login
2. **Data Processing**: Real-time data processing capabilities
3. **Reporting**: Generate comprehensive business reports
4. **Integration**: Connect with existing enterprise systems

Non-Functional Requirements

1. **Performance**: Response times under 2 seconds
2. **Scalability**: Support 1000+ concurrent users
3. **Security**: Meet enterprise security standards
4. **Availability**: 99.9% uptime requirement

Stakeholders

- Project Manager: Sarah Johnson
- Business Analyst: Mike Chen
- Technical Lead: David Wilson
- QA Manager: Lisa Brown

Success Criteria

The project will be considered successful when:

- All functional requirements are implemented
- Performance targets are met
- User acceptance criteria are satisfied
- Security compliance is achieved

This document serves as a foundation for demonstrating the ADPA hub features in Microsoft Word.

=== HUB-FEATURES-DEMO.MD (other) ===
Path: ADPA\demo\hub-features-demo.md
Relevance Score: 80

🎯 ADPA Hub Features Demo Guide

🚀 Live Demonstration: New Hub System in Word

📋 Prerequisites

✅ ADPA add-in loaded in Microsoft Word
✅ Development server running on localhost:3000
✅ Sample document with requirements content

🎮 Demo Script: Testing the 4 Hub Commands

Step 1: Load the ADPA Add-in

1. Open Microsoft Word
2. Go to **Insert** → **My Add-ins** → **ADPA**
3. Look for the **4 new hub buttons** in the ADPA ribbon:
    - 📄 **Document Conversion**
    - 📊 **Smart Diagrams**
    - 🤖 **AI Intelligence**
    - 👥 **Collaboration**

Step 2: Test Smart Diagrams Hub (Phase 3 Featured!)

🎯 **This showcases our Phase 3 Interactive Timeline feature prominently!**

1. **Click "Smart Diagrams" button**

    - **Expected**: Menu appears with "Interactive Timeline" highlighted as featured action
    - **Result**: Interactive Timeline executes immediately

2. **Interactive Timeline Features to Test**:

- ✅ **Click Events**: Click on timeline events to see detailed information
    - ✅ **Zoom Controls**: Use zoom in/out to focus on time periods
    - ✅ **Drag & Drop**: Drag events to reschedule them
    - ✅ **Real-time Updates**: Changes reflect immediately

3. **Access Other Diagram Features**:

    - Interactive Gantt Chart (Phase 3)
    - Enable Interactive Mode (Phase 3)
    - AI Smart Diagrams
    - Basic Diagram Generation
    - Custom Template Builder

Step 3: Test Document Conversion Hub

🎯 **Adobe PDF Generation prominently featured!**

1. **Click "Document Conversion" button**

    - **Expected**: Menu with Adobe PDF Generation highlighted
    - **Featured Action**: Professional PDF with ADPA templates

2. **Test Adobe Integration**:

    - ✅ Generate PDF with professional formatting
    - ✅ InDesign layout creation for print-ready documents
    - ✅ Multi-format package generation

3. **Access Other Conversion Features**:

    - Project Charter documents
    - Technical Specifications
    - B
      ... [truncated]

=== DATA-ARCHITECTURE-QUALITY-CHECKLIST.MD (development) ===
Path: to process\DATA-ARCHITECTURE-QUALITY-CHECKLIST.md
Relevance Score: 68

Checklist: Implementing the Data Architecture & Quality Document

This checklist outlines the tasks required to add the **Data Architecture & Quality** document type to the ADPA Document Generator.

---

1. Create Template and Processor Files

- ☑ Create the template file: `src/modules/documentTemplates/dmbok/DataArchitectureQualityTemplate.ts`

- ☑ Implement the `DataArchitectureQualityTemplate` function that defines the document structure.
- ☑ Create the processor file: `src/modules/documentTemplates/dmbok/DataArchitectureQualityProcessor.ts`
- ☑ Implement the `DataArchitectureQualityProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json`.
- ☑ Add a new entry for `data-architecture-quality`.
- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, such as `data-architecture-modeling-guide` and `data-quality-management-plan`.
- ☑ Assign a `priority` for generation order (suggested: 19).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts`.
- ☑ Add a new task object to the `GENERATION_TASKS` array for `data-architecture-quality`.
- ☑ Ensure the following fields are correctly filled out:
    - `key`: 'data-architecture-quality'
    - `name`: 'Data Architecture & Quality'
    - `category`: 'dmbok'
    - `func`: 'generateDataArchitectureQuality'
    - `emoji`: '🏗️'
    - `priority`: 19
    - `pmbokRef`: 'DMBOK: Data Architecture & Quality'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts`.
- ☑ Add a new entry to the `DOCUMENT_CONFIG` object for `data-architecture-quality`.
- ☑ Specify the following properties:
    - `title`: 'Data Architecture & Quality'
    - `filename`: 'dmbok/data-architecture-quality.md'
    - `category`: DOCUMENT_CATEGORIES.DMBOK
    - `description`: 'Defines

      ... [truncated]

=== BUSINESS ANALYSIS BODY OF KNOWLEDGE INTRODUCTION CHECKLIST.MD (primary) ===
Path: to process\BUSINESS ANALYSIS BODY OF KNOWLEDGE INTRODUCTION CHECKLIST.md
Relevance Score: 59

Checklist: Implementing the INTRODUCTION BUSINESS ANALYSIS BODY OF KNOWLEDGE

Document

This checklist outlines the tasks required to add the **INTRODUCTION BUSINESS ANALYSIS BODY OF KNOWLEDGE** document type to the ADPA Document Generator.

---

1. Create Template and Processor Files

- ☑ Create the template file: `src/modules/documentTemplates/babok/IntroductionBusinessAnalysisBodyOfKnowledgeTemplate.ts`
- ☑ Implement the `IntroductionBusinessAnalysisBodyOfKnowledgeTemplate` function that defines the document structure and content.
- ☑ Create the processor file: `src/modules/documentTemplates/babok/IntroductionBusinessAnalysisBodyOfKnowledgeProcessor.ts`
- ☑ Implement the `IntroductionBusinessAnalysisBodyOfKnowledgeProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json`.
- ☑ Add a new entry for `introduction-business-analysis-body-of-knowledge`.
- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, if any (e.g., other BABOK summary or index documents).
- ☑ Assign a `priority` for generation order (suggested: 1, as an introduction).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts`.
- ☑ Add a new task object to the `GENERATION_TASKS` array for `introduction-business-analysis-body-of-knowledge`.
- ☑ Ensure the following fields are correctly filled out:
  - ☑ `key`: 'introduction-business-analysis-body-of-knowledge'
  - ☑ `name`: 'Introduction Business Analysis Body of Knowledge'
  - ☑ `category`: 'babok'
  - ☑ `func`: 'generateIntroductionBusinessAnalysisBodyOfKnowledge'
  - ☑ `emoji`: '📘'
  - ☑ `priority`: 1
  - ☑ `babokRef`: 'BABOK: Introduction'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts`.
- ☑ Add a new entry to the `DOCUMENT_CONFIG` o
  ... [truncated]

=== DATA MANAGEMENT BODY OF KNOWLEDGE INTRODUCTION CHECKLIST.MD (primary) ===
Path: to process\DATA MANAGEMENT BODY OF KNOWLEDGE INTRODUCTION CHECKLIST.md

Relevance Score: 56

Checklist: Implementing the INTRODUCTION DATA MANAGEMENT BODY OF KNOWLEDGE Document

This checklist outlines the tasks required to add the **INTRODUCTION DATA MANAGEMENT BODY OF KNOWLEDGE** document type to the ADPA Document Generator.

---

1. Create Template and Processor Files

- ☑ Create the template file: `src/modules/documentTemplates/dmbok/IntroductionDataManagementBodyOfKnowledgeTemplate.ts`
- ☑ Implement the `IntroductionDataManagementBodyOfKnowledgeTemplate` function that defines the document structure and content.
- ☑ Create the processor file: `src/modules/documentTemplates/dmbok/IntroductionDataManagementBodyOfKnowledgeProcessor.ts`
- ☑ Implement the `IntroductionDataManagementBodyOfKnowledgeProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json`.
- ☑ Add a new entry for `introduction-data-management-body-of-knowledge`.
- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, if any (e.g., other DMBOK summary or index documents).
- ☑ Assign a `priority` for generation order (suggested: 1, as an introduction).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts`.
- ☑ Add a new task object to the `GENERATION_TASKS` array for `introduction-data-management-body-of-knowledge`.
- ☑ Ensure the following fields are correctly filled out:
    - `key`: 'introduction-data-management-body-of-knowledge'
    - `name`: 'Introduction Data Management Body of Knowledge'
    - `category`: 'dmbok'
    - `func`: 'generateIntroductionDataManagementBodyOfKnowledge'
    - `emoji`: '📚'
    - `priority`: 1
    - `pmbokRef`: 'DMBOK: Introduction'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts`.
- ☑ Add a new entry to the `DOCUMENT_CONFIG` object for `introduction-data-management-body-of-kn
  ... [truncated]

=== BUSINESS-INTELLIGENCE-STRATEGY-CHECKLIST.MD (other) ===
Path: to process\BUSINESS-INTELLIGENCE-STRATEGY-CHECKLIST.md
Relevance Score: 48

Checklist: Implementing the Business Intelligence & Analytics Strategy

This checklist outlines the tasks required to add the **Business Intelligence & Analytics Strategy** document type to the ADPA Document Generator.

---

1. Create Template and Processor Files

- ☑ Create the template file: `src/modules/documentTemplates/dmbok/BusinessIntelligenceStrategyTemplate.ts`
- ☑ Implement the `BusinessIntelligenceStrategyTemplate` class with a `buildPrompt()` method that defines the document structure.
- ☑ Create the processor file: `src/modules/documentTemplates/dmbok/BusinessIntelligenceStrategyProcessor.ts`
- ☑ Implement the `BusinessIntelligenceStrategyProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json`.
- ☑ Add a new entry for `business-intelligence-strategy`.
- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, such as `data-architecture-modeling-guide` and `data-governance-framework`.
- ☑ Assign a `priority` for generation order (suggested: 16).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts`.
- ☑ Add a new task object to the `GENERATION_TASKS` array for `business-intelligence-strategy`.
- ☑ Ensure the following fields are correctly filled out:
  - ☑ `key`: 'business-intelligence-strategy'
  - ☑ `name`: 'Business Intelligence & Analytics Strategy'
  - ☑ `category`: 'dmbok'
  - ☑ `func`: 'generateBusinessIntelligenceStrategy'
  - ☑ `emoji`: '📊'

    - ☑ `priority`: 16
    - ☑ `pmbokRef`: 'DMBOK: Business Intelligence & Analytics'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts`.
- ☑ Add a new entry to the `DOCUMENT_CONFIG` object for `business-intelligence-strategy`.
- ☑ Specify the following properties:
    - ☑ `title`: 'Business Intelligence & Analytics Strategy'
    -

    ... [truncated]

=== DATA-MODELING-STANDARDS-CHECKLIST.MD (other) ===
Path: to process\DATA-MODELING-STANDARDS-CHECKLIST.md
Relevance Score: 48

Checklist: Implementing the Data Modeling Standards Guide

This checklist outlines the tasks required to add the **Data Modeling Standards Guide** document type to the ADPA Document Generator.

- ☑ Create the template file: `src/modules/documentTemplates/dmbok/DataModelingStandardsTemplate.ts`
- ☑ Implement the `DataModelingStandardsTemplate` (now as a function) that defines the document structure.
- ☑ Create the processor file: `src/modules/documentTemplates/dmbok/DataModelingStandardsProcessor.ts`
- ☑ Implement the `DataModelingStandardsProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json`.
- ☑ Add a new entry for `data-modeling-standards`.
- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, such as `data-architecture-modeling-guide`.
- ☑ Assign a `priority` for generation order (suggested: 17).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts`.
- ☑ Add a new task object to the `GENERATION_TASKS` array for `data-modeling-standards`.
- ☑ Ensure the following fields are correctly filled out:
    - `key`: 'data-modeling-standards'

    - `name` : 'Data Modeling Standards Guide'
    - `category` : 'dmbok'
    - `func` : 'generateDataModelingStandards'
    - `emoji` : '📐'
    - `priority` : 17
    - `pmbokRef` : 'DMBOK: Data Modeling & Design'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts` .
- ☑ Add a new entry to the `DOCUMENT_CONFIG` object for `data-modeling-standards` .
- ☑ Specify the following properties:
    - `title` : 'Data Modeling Standards Guide'
    - `filename` : 'dmbok/data-modeling-standards.md'
    - `category` : DOCUMENT_CATEGORIES.DMBOK
    - `description` : 'Comprehensive guide to data modeling standards, conventions, and best practices.'
    - `generate
      ... [truncated]

=== ENTERPRISE-DATA-DICTIONARY-CHECKLIST.MD (other) ===
Path: to process\ENTERPRISE-DATA-DICTIONARY-CHECKLIST.md
Relevance Score: 45

Checklist: Implementing the Enterprise Data Dictionary

This checklist outlines the tasks required to add the **Enterprise Data Dictionary** document type to the ADPA Document Generator.

---

1. Create Template and Processor Files

- ☑ Create the template file: `src/modules/documentTemplates/dmbok/EnterpriseDataDictionaryTemplate.ts`
- ☑ Implement the `EnterpriseDataDictionaryTemplate` class with a `buildPrompt()` method that defines the document structure.
- ☑ Create the processor file: `src/modules/documentTemplates/dmbok/EnterpriseDataDictionaryProcessor.ts`
- ☑ Implement the `EnterpriseDataDictionaryProcessor` class, ensuring it uses the template and the `AIProcessor` to generate the document content.

2. Register the Processor

- ☑ Open `src/modules/documentGenerator/processor-config.json` .
- ☑ Add a new entry for `enterprise-data-dictionary` .

- ☑ Define the `module` path pointing to the new processor class.
- ☑ List dependencies, such as `data-modeling-standards` and `metadata-management-framework` .
- ☑ Assign a `priority` for generation order (suggested: 18).

3. Add a Generation Task

- ☑ Open `src/modules/documentGenerator/generationTasks.ts` .
- ☑ Add a new task object to the `GENERATION_TASKS` array for `enterprise-data-dictionary` .
- ☑ Ensure the following fields are correctly filled out:
  - ☑ `key` : 'enterprise-data-dictionary'
  - ☑ `name` : 'Enterprise Data Dictionary'
  - ☑ `category` : 'dmbok'
  - ☑ `func` : 'generateEnterpriseDataDictionary'
  - ☑ `emoji` : '📚'
  - ☑ `priority` : 18
  - ☑ `pmbokRef` : 'DMBOK: Metadata Management'

4. Add File Manager Configuration

- ☑ Open `src/modules/fileManager.ts` .
- ☑ Add a new entry to the `DOCUMENT_CONFIG` object for `enterprise-data-dictionary` .
- ☑ Specify the following properties:
  - ☑ `title` : 'Enterprise Data Dictionary'
  - ☑ `filename` : 'dmbok/enterprise-data-dictionary.md'
  - ☑ `category` : DOCUMENT_CATEGORIES.DMBOK
  - ☑ `descriptio

    ... [truncated]

=== BABOK CHECKLISTS.MD (other) ===
Path: to process\BABOK CHECKLISTS.md
Relevance Score: 44

BABOK Document Checklists

This file provides a checklist for each core BABOK knowledge area/document. Use these to track implementation, review, and validation for each BABOK deliverable in your project.

---

1. Business Analysis Planning & Monitoring

- ☐ Define the purpose and scope of the document
- ☐ Identify stakeholders and their roles
- ☐ Document planning approach and deliverables
- ☐ Establish monitoring and reporting mechanisms
- ☐ Review and validate with stakeholders
- ☐ Finalize and approve the document

2. Elicitation & Collaboration

- □ Prepare for elicitation activities
- □ Conduct elicitation sessions (interviews, workshops, etc.)
- □ Document elicitation results
- □ Collaborate with stakeholders for feedback
- □ Confirm and validate requirements
- □ Finalize elicitation documentation

3. Requirements Life Cycle Management

- □ Define requirements traceability approach
- □ Establish requirements change management process
- □ Maintain requirements documentation
- □ Track requirements status and approvals
- □ Validate requirements with stakeholders
- □ Archive or retire obsolete requirements

4. Strategy Analysis

- □ Identify business needs and drivers
- □ Assess current state and define future state
- □ Analyze gaps and recommend solutions
- □ Define business case and value proposition
- □ Review strategy with stakeholders
- □ Finalize and approve strategy documentation

5. Requirements Analysis & Design Definition

- □ Structure and organize requirements
- □ Specify and model requirements and designs
- □ Validate and verify requirements
- □ Define acceptance criteria
- □ Review with stakeholders
- □ Finalize analysis and design documentation

6. Solution Evaluation

- □ Define evaluation criteria and metrics
- □ Assess solution performance and value
- □ Identify limitations and improvement areas
- □ Collect feedback
  ... [truncated]

=== BABOK DOCUMENT CREATION CHECKLISTS.MD (other) ===
Path: to process\BABOK DOCUMENT CREATION CHECKLISTS.md
Relevance Score: 35

BABOK Document Creation & Implementation Checklists

This file provides a standardized checklist for the creation and implementation of each BABOK document/knowledge area in the ADPA Document Generator. Each checklist follows the same format as used for other document types in this project.

- ☑ Create the template file: `src/modules/documentTemplates/babok/BusinessAnalysisPlanningAndMonitoringTemplate.ts`
- ☑ Implement the template as a class with a `generateContent(context: ProjectContext): string` method for document structure and content. Use `ProjectContext` for context typing.
- ☑ Create the processor file: `src/modules/documentTemplates/babok/BusinessAnalysisPlanningAndMonitoringProcessor.ts`
- ☑ Implement the processor class to:
  - ☑ Implement the `DocumentProcessor` interface and return a `DocumentOutput`.
  - ☑ Use composition (not inheritance) with AIProcessor, accessed via `getAIProcessor()`.
  - ☑ Accept a typed `ProjectContext` for the context parameter.
  - ☑ Use the template as a structure reference for AI-enhanced content, with fallback to the template if AI fails.
  - ☑ Include robust error handling and output validation.
  - ☑ Add clear JSDoc comments for maintainability.
- ☑ Register the processor in `processor-config.json`.
- ☑ Add a generation task in `generationTasks.ts`.
- ☑ Add an entry to `DOCUMENT_CONFIG` in `fileManager.ts`.
- ☑ Build the project: `npm run build`
- ☑ Run the generator for this document with verbose output.
- ☑ Verify the processor loads without errors.
- ☑ Confirm the document is listed: `node dist/cli.js list-templates`
- ☑ Inspect the generated file for content and formatting.
- ☑ Update `README.md` to include this document and usage example.
- ☑ Create the template file: `src/modules/documentTemplates/babok/ElicitationAndCollaborationTemplate.ts`
- ☑ Implement the template as a class with a `generateContent(context: ProjectContext
  ... [truncated]

Introduction

Overview of deployment architecture and deployment strategy.

Deployment Architecture

High-level architecture of the deployment environment and infrastructure.

Environment Setup

Setup and configuration of deployment environments.

Infrastructure Requirements

Hardware, software, and network requirements for deployment.

Deployment Strategies

Different deployment strategies including blue-green, canary, and rolling deployments.

Configuration Management

Management of configuration files and environment-specific settings.

Security Considerations

Security best practices for deployment including access control and data protection.

Monitoring and Health Checks

Implementation of monitoring, health checks, and alerting systems.

Backup and Recovery

Backup strategies and disaster recovery procedures.

Scaling Procedures

Procedures for scaling applications horizontally and vertically.

Maintenance and Updates

Procedures for performing maintenance and applying updates.

```
# ADPA Enterprise Framework Deployment Guidance

This document provides comprehensive, project-specific deployment procedures and recommendations for
```

~~~
---

## 1. Deployment Procedures

### 1.1. Prerequisites

- **Node.js**: v18.0.0 or higher ([Download](https://nodejs.org/))
- **npm**: v8.x or higher (bundled with Node.js)
- **Git**: For source deployments
- **Supported OS**: Linux (recommended), Windows, macOS
- **Optional**: Docker (when containerization is available), Redis

---

### 1.2. Deployment Options Overview

- **NPM Global CLI**: For local or developer use
- **From Source**: For CI/CD, extensibility, and enterprise control
- **Docker Container**: (Coming soon) For containerized, orchestrated deployments
- **Kubernetes**: (Planned) For large-scale, production-grade orchestration

---

### 1.3. NPM Global Installation (Recommended for CLI/Quick API)

```bash
npm install -g adpa-enterprise-framework-automation
~~~

- **CLI Usage**: `adpa generate ...`
- **API Server**: `adpa-api` (starts Express.js server)

1.4. Deployment From Source (Enterprise/Custom)

```
 git clone https://github.com/mdresch/requirements-gathering-agent.git
cd requirements-gathering-agent
npm install
npm run build
```

Start API Server

```
 npm run api:start
# API available at http://localhost:3000
```

Start Admin Web Portal

```
 npm run admin:setup
npm run admin:serve
# Access at http://localhost:3001
```

Optional: Run in Development Mode

```
npm run dev
```

1.5. Docker Deployment (When Available)

```
docker pull adpa/enterprise-framework:latest
docker run -d -p 3000:3000 --env-file .env adpa/enterprise-framework:latest
```

- **Mount volumes** for persistent storage or configuration as needed.
- **Kubernetes Helm charts** and templates are planned for Q2/Q3 2025.

1.6. Multi-Node/Production Deployment

- Use **process managers** (e.g., PM2, systemd, supervisor) to ensure uptime.
- Behind **reverse proxies** (e.g., NGINX) for SSL termination, load balancing, and API gateway features.
- **Horizontal scaling**: Run multiple API/admin instances behind a load balancer.
- **Stateless API**: Use externalized session stores (Redis/DB) for scalability.

---

2. Environment-Specific Configuration

2.1. Environment Variables

Copy and configure your `.env`:

```
cp .env .env.local
```

Minimum Required Variables

- `NODE_ENV` = `development`, `staging`, or `production`
- `PORT` (default 3000 for API, 3001 for admin)
- `JWT_SECRET` (use a strong, unique value)
- **AI Provider settings** (see below)
- **Database settings** (if switching from local JSON to MongoDB/SQL)

2.2. AI Provider Configuration

Choose and set one primary AI provider, and configure fallbacks:

Google AI (Recommended - Free Tier)

```
CURRENT_PROVIDER=google-ai
GOOGLE_AI_API_KEY=xxxxx
GOOGLE_AI_MODEL=gemini-1.5-flash
```

GitHub AI

```
CURRENT_PROVIDER=github-ai
GITHUB_TOKEN=xxxxx
GITHUB_ENDPOINT=https://models.github.ai/inference/
REQUIREMENTS_AGENT_MODEL=gpt-4o-mini
```

Azure OpenAI (Enterprise)

```
CURRENT_PROVIDER=azure-openai-key
AZURE_OPENAI_ENDPOINT=https://your-resource.openai.azure.com/
AZURE_OPENAI_API_KEY=xxxxx
AZURE_OPENAI_DEPLOYMENT_NAME=your-deployment
AZURE_OPENAI_API_VERSION=2024-02-15-preview
```

Ollama (Local)

```
CURRENT_PROVIDER=ollama
OLLAMA_ENDPOINT=http://localhost:11434/api
OLLAMA_MODEL=deepseek-coder:latest
```

> ***Tip:*** *For enterprise, use **Azure OpenAI** with Entra ID for maximum control and compliance.*

2.3. Integration Credentials

- **Confluence**: OAuth2 client ID/secret, base URL
- **SharePoint**: Azure app registration, OAuth2, tenant ID
- **Adobe PDF Services**: API key, client ID/secret

Store all credentials in secure secrets management (e.g., Azure Key Vault, AWS Secrets Manager) in production.

2.4. API & Admin Portal URLs

- Expose via HTTPS with trusted certificates.
- Configure public and internal DNS as needed.
- Set CORS origins (`CORS_ORIGIN`) for the API as required.

2.5. Database Backend

- Default is JSON-based config; for production, use **MongoDB** (via Mongoose) or another enterprise-ready DB.
- Configure connection strings in `.env`.

---

3. Security Best Practices

3.1. API Security

- **HTTPS everywhere**: Terminate at load balancer or reverse proxy.
- **JWT-based authentication**: Use strong secrets, short expiry, and refresh tokens.
- **Role-based access control (RBAC)**: Enforce using Express middleware.
- **Input validation**: All endpoints use `joi` / `express-validator` for sanitization.
- **Rate limiting**: Enabled via `express-rate-limit` (tune thresholds per deployment).
- **Helmet**: Set HTTP security headers.
- **Audit logging**: Use `winston` / `morgan` for request and error logs.

3.2. Data Security & Compliance

- **At-rest encryption**: Use encrypted volumes for all persistent data.
- **In-transit encryption**: TLS 1.2+ for all connections.
- **Secrets management**: Never commit secrets/keys to source; use environment variables and secret stores.
- **Compliance**: Ensure configurations and integrations adhere to GDPR, SOX, PCI DSS, etc.
- **User management**: Integrate with enterprise SSO (OAuth2, SAML, AD) for identity and access.

3.3. Dependency & Code Security

- Run `npm audit` regularly.
- Monitor for vulnerable packages.
- Use **dependabot** or similar for automated dependency updates.
- Enforce code scanning in CI/CD.

3.4. Admin Portal Security

- Restrict access to trusted internal networks or VPN.
- Enable 2FA/SSO for admin users.
- Monitor and log all admin actions.

---

4. Disaster Recovery & Backup

4.1. Backup Strategy

- **Configuration & Documents**: Regularly back up `docs/`, `generated-documents/`, and configuration files.
- **Database**: Schedule automated backups of MongoDB/SQL databases.
- **AI Provider Configs**: Store provider credentials and models in secure, versioned vaults.
- **Cloud Storage**: Use versioned cloud storage for all generated output.

4.2. Recovery Procedures

- **Automated restore scripts**: For DB and document storage.
- **Immutable backups**: Use write-once storage for critical compliance data.
- **Disaster simulation**: Test restores quarterly.

4.3. High Availability

- Deploy multiple API/admin nodes across availability zones.
- Use managed DB with HA/failover features.
- Implement health checks and auto-restart with process managers (PM2, Kubernetes, etc).

4.4. Monitoring & Alerting

- Integrate with monitoring systems (Prometheus, Grafana, Azure Monitor, etc.).
- Set up alerts for downtime, error spikes, and abnormal resource usage.

---

5. Performance & Scalability Considerations

5.1. Scaling

- **Stateless API**: Scale horizontally with load balancers.
- **Session and cache storage**: Use Redis for scalable session/caching.
- **Microservices**: Modularize services for future scaling (planned for future releases).

5.2. Tuning

- **Node.js process**: Tune memory limits (`--max-old-space-size`), garbage collection.
- **Database indexing**: For MongoDB/SQL, ensure indexes on document and user data.
- **Document generation**: Use job queues (Bull, RabbitMQ) for expensive operations in high-load scenarios.

5.3. Caching

- Enable Redis-based caching for AI responses, document templates, and metadata.
- Cache static assets via CDN for admin portal.

5.4. Load Testing

- Use `npm run test:performance` to verify document generation and API throughput under load.
- Run integration and provider-specific tests (`npm run test:azure` , etc.) before production cut-over.

5.5. Logging & Analytics

- Centralize logs (e.g., ELK, Azure Log Analytics).
- Monitor API latency, error rates, and document generation durations.

---

6. Example Production Topology

```
 [Users/Clients]
     |
[Reverse Proxy / Load Balancer]
     |             |
[API Node 1]  [API Node 2]  ...   [Admin Portal Node(s)]
     |             |
   [Redis / DB / File Storage]
     |
[External Integrations (AI, Confluence, SharePoint, Adobe)]
```

---

7. Additional Recommendations

- **Regularly update dependencies** and monitor for breaking changes in AI provider APIs.
- **Document all local modifications** for upgrades.
- **Use feature flags** for experimental features in production.
- **Review roadmap** for Docker/K8s support and plan for migrations.

---

8. References

- Project Wiki & Docs
- Issue Tracker
- Enterprise Support

This guidance is tailored for the ADPA platform version 3.2.0 and later, and should be reviewed for each major release.

```

```