

Feature Specification: Desktop Video Converter

Feature Branch: 001-desktop-video-converter

Created: September 16, 2025

Status: Draft

Input: User description: "Desktop Video Converter - A simple, single-purpose Windows desktop application that converts any video file into a web-optimized MP4 using Electron framework. Features drag-and-drop file selection, real-time progress tracking, and packages as a portable .exe file."

Execution Flow (main)

1. Parse user description from Input
 - Feature parsed: Desktop video converter application
2. Extract key concepts from description
 - Actors: Windows desktop users, Actions: convert videos to MP4, Data: video files, Constraints: web-optimized output, portable .exe
3. For each unclear aspect:
 - All key aspects clearly defined in PRD
4. Fill User Scenarios & Testing section
 - Clear user flow: select file → choose destination → convert → save
5. Generate Functional Requirements
 - All requirements testable and unambiguous
6. Identify Key Entities
 - Video files, conversion jobs, user preferences
7. Run Review Checklist
 - No clarifications needed, implementation details minimal
8. Return: SUCCESS (spec ready for planning)

⚡ Quick Guidelines

- ☒ Focus on WHAT users need and WHY
- ☒ Avoid HOW to implement (no tech stack, APIs, code structure)
- ☒ Written for business stakeholders, not developers

User Scenarios & Testing (*mandatory*)

Primary User Story

A home user has a video file they want to optimize for web use. They open the Desktop Video Converter application, select their video file either by clicking "Select Video File" or dragging it onto the window, confirm the destination (defaulting to Desktop with "_converted" suffix), click "Convert", watch the progress bar, and receive their optimized MP4 file in the chosen location.

Acceptance Scenarios

1. **Given** the application is launched, **When** user clicks "Select Video File" and chooses a valid video, **Then** the filename displays and "Convert" button becomes active
2. **Given** a video file is selected, **When** user clicks "Convert", **Then** progress bar appears and conversion begins with real-time updates
3. **Given** conversion is in progress, **When** user clicks "Cancel", **Then** process stops immediately and temporary files are cleaned up
4. **Given** conversion completes successfully, **When** user sees success message, **Then** "Open Folder" button opens destination directory in Windows Explorer
5. **Given** no video file selected, **When** user attempts any action, **Then** "Convert" button remains disabled until file selection

Edge Cases

- What happens when selected file is not a valid video format? System displays error message and allows new selection
- How does system handle insufficient disk space? Conversion fails gracefully with clear error message
- What if user selects read-only destination? Error displayed with request to choose different location
- How does system behave if conversion is interrupted by system shutdown? Temporary files cleaned on next application start

Requirements (*mandatory*)

Functional Requirements

- **FR-001:** System **MUST** allow users to select video files from local Windows file system via button click or drag-and-drop
- **FR-002:** System **MUST** display selected video filename clearly in the interface
- **FR-003:** System **MUST** provide destination path selection with default to user's Desktop directory
- **FR-004:** System **MUST** automatically generate output filename with "_converted" suffix while preserving original name
- **FR-005:** System **MUST** enable "Convert" button only after valid video file selection
- **FR-006:** System **MUST** convert selected video to web-optimized MP4 format with best quality-to-size ratio
- **FR-007:** System **MUST** display real-time progress indicator during conversion process
- **FR-008:** System **MUST** disable all controls except "Cancel" during active conversion
- **FR-009:** System **MUST** provide immediate conversion cancellation with complete cleanup of temporary files
- **FR-010:** System **MUST** save converted file to user-specified destination path
- **FR-011:** System **MUST** display success message and provide "Open Folder" functionality upon completion
- **FR-012:** System **MUST** include accessible "About" section displaying application and developer information
- **FR-013:** System **MUST** package as single portable .exe file requiring no installation
- **FR-014:** System **MUST** run on Windows 10 and Windows 11 operating systems
- **FR-015:** System **MUST** maintain clean, intuitive interface suitable for non-technical users
- **FR-016:** System **MUST** support keyboard-only navigation for accessibility compliance (WCAG AA)

Key Entities *(include if feature involves data)*

- **Video File:** Represents source media file selected by user, includes path, name, format, and validation status
 - **Conversion Job:** Represents active conversion process, includes progress percentage, status (running/cancelled/completed), source and destination paths
 - **Application State:** Represents current UI state including selected file, destination path, conversion status, and control availability
 - **User Preferences:** Represents application settings including default destination directory and window positioning
-

Review & Acceptance Checklist

GATE: Automated checks run during main() execution

Content Quality

- ☒ No implementation details (languages, frameworks, APIs)
- ☒ Focused on user value and business needs
- ☒ Written for non-technical stakeholders
- ☒ All mandatory sections completed

Requirement Completeness

- ☒ No [NEEDS CLARIFICATION] markers remain
 - ☒ Requirements are testable and unambiguous
 - ☒ Success criteria are measurable
 - ☒ Scope is clearly bounded
 - ☒ Dependencies and assumptions identified
-

Execution Status

Updated by main() during processing

- ☒ User description parsed
- ☒ Key concepts extracted
- ☒ Ambiguities marked
- ☒ User scenarios defined
- ☒ Requirements generated
- ☒ Entities identified
- ☒ Review checklist passed