Tasks: Desktop Video Converter

Input: Design documents from `/specs/001-desktop-video-converter/`
Prerequisites: plan.md (required), research.md, data-model.md, contracts/

Execution Flow (main)

```
1. Load plan.md from feature directory
   → ☑ Found: React + Electron + Vite with shadcn/ui and FFmpeg
   → Extract: TypeScript, electron-vite, fluent-ffmpeg, shadcn/ui
2. Load optional design documents:
   → data-model.md: Extract entities → model tasks
   → contracts/: ipc-contracts.md, api-contracts.md → contract test tasks
   → research.md: Extract decisions → setup tasks
3. Generate tasks by category:
   → Setup: project init, dependencies, linting
   → Tests: contract tests, integration tests
   → Core: models, services, IPC handlers
   → Integration: FFmpeg, UI components, packaging
   → Polish: unit tests, performance, docs
4. Apply task rules:
   → Different files = mark [P] for parallel
   → Same file = sequential (no [P])
   → Tests before implementation (TDD)
5. Number tasks sequentially (T001, T002...)
6. Generate dependency graph
7. Create parallel execution examples
8. Validate task completeness:
   → All contracts have tests? ☑
   → All entities have models? ☑
   → All IPC handlers implemented? ☑
9. Return: SUCCESS (tasks ready for execution)
```

Format: `[ID] [P?] Description`

- **[P]**: Can run in parallel (different files, no dependencies)
- Include exact file paths in descriptions

Path Conventions

Based on plan.md structure: Electron app with main/, renderer/, preload/, shared/

- **Main process**: main/services/, main/ipc-handlers/
- **Renderer process**: src/components/, src/hooks/, src/lib/
- **Shared types**: shared/types/
- **Tests**: tests/unit/, tests/integration/, tests/e2e/

Phase 3.1: Setup

- [ ] T001 Create Electron + Vite project structure per implementation plan with electron/, src/, shared/, main/ directories
- [ ] T002 Initialize TypeScript project with electron-vite, React 18, Vite 5, Electron 27 dependencies from research.md
- [ ] T003 [P] Configure ESLint and Prettier for TypeScript + React + Electron project
- [ ] T004 [P] Configure Tailwind CSS and install shadcn/ui components (button, card, progress, badge, alert-dialog)
- [ ] T005 [P] Setup Vitest for unit testing and Playwright for E2E testing per research.md
- [ ] T006 [P] Install FFmpeg dependencies (fluent-ffmpeg, ffmpeg-static) and configure static binary paths

Phase 3.2: Tests First (TDD) ⚠ MUST COMPLETE BEFORE 3.3

CRITICAL: These tests MUST be written and MUST FAIL before ANY implementation

Contract Tests

- [ ] T007 [P] IPC contract test file:select in tests/contracts/test-file-operations.spec.ts
- [ ] T008 [P] IPC contract test file:save-location in tests/contracts/test-file-operations.spec.ts
- [ ] T009 [P] IPC contract test file:validate in tests/contracts/test-file-validation.spec.ts
- [ ] T010 [P] IPC contract test conversion:start in tests/contracts/test-conversion-operations.spec.ts
- [ ] T011 [P] IPC contract test conversion:cancel in tests/contracts/test-conversion-operations.spec.ts
- [ ] T012 [P] IPC contract test conversion:progress event in tests/contracts/test-conversion-progress.spec.ts
- [ ] T013 [P] IPC contract test app:get-preferences in tests/contracts/test-app-state.spec.ts
- [ ] T014 [P] IPC contract test app:set-preferences in tests/contracts/test-app-state.spec.ts
- [ ] T015 [P] IPC contract test system:show-in-explorer in tests/contracts/test-system-integration.spec.ts
- [ ] T016 [P] IPC contract test app:info and app:quit in tests/contracts/test-app-lifecycle.spec.ts

API Contract Tests

- [ ] T017 [P] FFmpeg service contract test validateFile in tests/contracts/test-ffmpeg-service.spec.ts
- [ ] T018 [P] FFmpeg service contract test convertVideo in tests/contracts/test-ffmpeg-service.spec.ts
- [ ] T019 [P] File service contract test in tests/contracts/test-file-service.spec.ts
- [ ] T020 [P] Preferences service contract test in tests/contracts/test-preferences-service.spec.ts

Integration Tests

- [ ] T021 [P] Integration test file selection workflow in tests/integration/test-file-selection.spec.ts
- [ ] T022 [P] Integration test video conversion workflow in tests/integration/test-video-conversion.spec.ts
- [ ] T023 [P] Integration test progress tracking workflow in tests/integration/test-progress-tracking.spec.ts
- [ ] T024 [P] Integration test error handling workflow in tests/integration/test-error-handling.spec.ts

- [ ] T025 [P] Integration test settings persistence workflow in tests/integration/test-settings-persistence.spec.ts

Phase 3.3: Core Implementation (ONLY after tests are failing)

Shared Types

- [ ] T026 [P] VideoFile and VideoMetadata types in shared/types/video-file.ts
- [ ] T027 [P] ConversionJob and ConversionProgress types in shared/types/conversion-job.ts
- [ ] T028 [P] ApplicationState and UserPreferences types in shared/types/application-state.ts
- [ ] T029 [P] AppSession and KeyboardShortcuts types in shared/types/app-session.ts
- [ ] T030 [P] IPC contract types export in shared/types/ipc-contracts.ts
- [ ] T031 [P] API contract types export in shared/types/api-contracts.ts

Main Process Services

- [ ] T032 [P] FFmpeg service implementation in main/services/ffmpeg-service.ts
- [ ] T033 [P] File service implementation in main/services/file-service.ts
- [ ] T034 [P] Preferences service implementation in main/services/preferences-service.ts
- [ ] T035 [P] History service implementation in main/services/history-service.ts
- [ ] T036 [P] Notification service implementation in main/services/notification-service.ts
- [ ] T037 [P] Error handler service implementation in main/services/error-handler-service.ts
- [ ] T038 Service container and dependency injection in main/services/service-container.ts

IPC Handlers

- [ ] T039 [P] File operations IPC handlers in main/ipc-handlers/file-handlers.ts
- [ ] T040 [P] Conversion operations IPC handlers in main/ipc-handlers/conversion-handlers.ts
- [ ] T041 [P] Application state IPC handlers in main/ipc-handlers/app-handlers.ts
- [ ] T042 [P] System integration IPC handlers in main/ipc-handlers/system-handlers.ts

Electron Main and Preload

- [ ] T043 Electron main process entry point in electron/main.ts
- [ ] T044 Context bridge setup in electron/preload.ts
- [ ] T045 Window management and application lifecycle in electron/window.ts

Phase 3.4: Renderer Implementation

UI Components (shadcn/ui based)

- [ ] T046 [P] FileDropZone component with drag-and-drop in src/components/FileDropZone.tsx
- [ ] T047 [P] ConversionPanel component with quality settings in src/components/ConversionPanel.tsx
- [ ] T048 [P] ProgressDisplay component with real-time updates in src/components/ProgressDisplay.tsx
- [ ] T049 [P] VideoFileInfo component for metadata display in src/components/VideoFileInfo.tsx
- [ ] T050 [P] ErrorDialog component for error handling in src/components/ErrorDialog.tsx

- [ ] T051 [P] SettingsDialog component for user preferences in src/components/SettingsDialog.tsx

React Hooks and State Management

- [ ] T052 [P] useVideoFile hook for file state management in src/hooks/useVideoFile.ts
- [ ] T053 [P] useConversion hook for conversion operations in src/hooks/useConversion.ts
- [ ] T054 [P] useProgress hook for progress tracking in src/hooks/useProgress.ts
- [ ] T055 [P] usePreferences hook for settings management in src/hooks/usePreferences.ts
- [ ] T056 [P] useElectronAPI hook for IPC communication in src/hooks/useElectronAPI.ts

Utilities and Helpers

- [ ] T057 [P] File validation utilities in src/lib/file-utils.ts
- [ ] T058 [P] Format conversion utilities in src/lib/format-utils.ts
- [ ] T059 [P] Progress calculation utilities in src/lib/progress-utils.ts
- [ ] T060 [P] Error formatting utilities in src/lib/error-utils.ts

Main App Components

- [ ] T061 Main App component with routing in src/App.tsx
- [ ] T062 React entry point and providers in src/main.tsx

Phase 3.5: Integration and Configuration

Build Configuration

- [ ] T063 [P] electron-vite configuration in electron.vite.config.ts
- [ ] T064 [P] Vite renderer configuration with shadcn/ui aliases in vite.config.ts
- [ ] T065 [P] TypeScript configuration for main process in electron/tsconfig.json
- [ ] T066 [P] TypeScript configuration for renderer in tsconfig.json

Packaging and Distribution

- [ ] T067 [P] electron-builder configuration for Windows portable exe in build/electron-builder.js
- [ ] T068 [P] Package.json scripts for dev, build, and package commands
- [ ] T069 [P] Static asset handling and icon setup in build/

Development Tools

- [ ] T070 [P] Electron development menu and debug tools setup
- [ ] T071 [P] Hot module replacement configuration for renderer
- [ ] T072 [P] Logging configuration with electron-log

Phase 3.6: Polish and Testing

Unit Tests

- [ ] T073 [P] Unit tests for FFmpeg service in tests/unit/services/ffmpeg-service.test.ts
- [ ] T074 [P] Unit tests for file utilities in tests/unit/lib/file-utils.test.ts

- [ ] T075 [P] Unit tests for React hooks in tests/unit/hooks/
- [ ] T076 [P] Unit tests for UI components in tests/unit/components/

End-to-End Tests

- [ ] T077 [P] E2E test complete conversion workflow in tests/e2e/conversion-workflow.spec.ts
- [ ] T078 [P] E2E test drag and drop functionality in tests/e2e/drag-drop.spec.ts
- [ ] T079 [P] E2E test settings and preferences in tests/e2e/settings.spec.ts
- [ ] T080 [P] E2E test error scenarios in tests/e2e/error-handling.spec.ts

Performance and Optimization

- [ ] T081 [P] Performance testing for large video files (>1GB)
- [ ] T082 [P] Memory usage optimization and monitoring
- [ ] T083 [P] Bundle size optimization for distribution

Documentation and Final Polish

- [ ] T084 [P] Update README.md with setup and usage instructions
- [ ] T085 [P] Create user manual in docs/user-guide.md
- [ ] T086 [P] Code cleanup and remove debug logging
- [ ] T087 Manual testing per quickstart.md scenarios

Dependencies

Setup Dependencies

- T001 → T002 → T003,T004,T005,T006 (sequential setup)

Test Dependencies

- T002-T006 (setup) → T007-T025 (all contract and integration tests)

Core Implementation Dependencies

- T007-T025 (tests) → T026-T031 (shared types) → T032-T042 (services and handlers)
- T032-T038 (services) → T039-T042 (IPC handlers)
- T039-T042 (IPC handlers) → T043-T045 (Electron main/preload)

Renderer Dependencies

- T030-T031 (shared types) → T046-T062 (renderer components)
- T044 (preload) → T056 (useElectronAPI hook)
- T052-T056 (hooks) → T046-T051 (UI components)
- T057-T060 (utilities) → T046-T062 (components)

Integration Dependencies

- T043-T045 (Electron) + T061-T062 (renderer) → T063-T072 (configuration)
- T063-T069 (build config) → T073-T087 (testing and polish)

Parallel Execution Examples

Phase 3.1 Setup (after T002)

```
# Launch T003-T006 together:
Task: "Configure ESLint and Prettier for TypeScript + React + Electron project"
Task: "Configure Tailwind CSS and install shadcn/ui components"
Task: "Setup Vitest for unit testing and Playwright for E2E testing"
Task: "Install FFmpeg dependencies and configure static binary paths"
```

Phase 3.2 Contract Tests (after setup)

```
# Launch T007-T016 together:
Task: "IPC contract test file:select in tests/contracts/test-file-
operations.spec.ts"
Task: "IPC contract test conversion:start in tests/contracts/test-conversion-
operations.spec.ts"
Task: "IPC contract test app:get-preferences in tests/contracts/test-app-
state.spec.ts"
# ... (all contract tests can run in parallel)
```

Phase 3.3 Shared Types (after tests fail)

```
# Launch T026-T031 together:
Task: "VideoFile and VideoMetadata types in shared/types/video-file.ts"
Task: "ConversionJob and ConversionProgress types in shared/types/conversion-
job.ts"
Task: "ApplicationState and UserPreferences types in shared/types/application-
state.ts"
# ... (all type definitions can run in parallel)
```

Phase 3.4 UI Components (after hooks)

```
# Launch T046-T051 together:
Task: "FileDropZone component with drag-and-drop in
src/components/FileDropZone.tsx"
Task: "ConversionPanel component with quality settings in
src/components/ConversionPanel.tsx"
Task: "ProgressDisplay component with real-time updates in
src/components/ProgressDisplay.tsx"
# ... (all UI components can run in parallel)
```

Notes

- [P] tasks = different files, no dependencies
- Verify tests fail before implementing (TDD enforced)
- Commit after each task completion
- Follow security patterns from research.md (context bridge isolation)
- Use shadcn/ui components consistently
- Implement FFmpeg progress tracking per research findings

Task Generation Rules Applied

1. **From IPC Contracts**:
    - file:select, file:save-location, file:validate → T007-T009
    - conversion:start, conversion:cancel, conversion:progress → T010-T012
    - app:get-preferences, app:set-preferences → T013-T014
    - system:show-in-explorer, app:info, app:quit → T015-T016
2. **From API Contracts**:
    - FFmpegService, FileService, PreferencesService → T017-T020
    - Service implementations → T032-T037
3. **From Data Model**:
    - VideoFile, ConversionJob, ApplicationState, AppSession → T026-T029
    - Type exports → T030-T031
4. **From User Stories (quickstart.md)**:
    - File selection workflow → T021
    - Video conversion workflow → T022
    - Progress tracking workflow → T023
    - Error handling workflow → T024
    - Settings persistence workflow → T025
5. **Ordering Applied**:
    - Setup → Tests → Models → Services → UI → Integration → Polish
    - Tests always before implementation (TDD)
    - Types before services, services before handlers

Validation Checklist

GATE: Checked by main() before returning

- ☑ All IPC contracts have corresponding tests (T007-T016)
- ☑ All API contracts have corresponding tests (T017-T020)
- ☑ All entities have model tasks (T026-T029)
- ☑ All tests come before implementation (T007-T025 before T026+)
- ☑ Parallel tasks truly independent (different files)
- ☑ Each task specifies exact file path

- [x] No task modifies same file as another [P] task
- [x] TDD pattern enforced (tests → implementation → polish)
- [x] All user workflows covered by integration tests
- [x] Build and packaging tasks included (T063-T069)
- [x] Performance and documentation tasks included (T081-T087)