The JextFrame class serves as the main application window for the Jext Java text editor.
It orchestrates menus, toolbars, editor panes, consoles, and plugin integration.
It also manages splash screens, workspace loading, and batch-mode file opening.
This constructor-centric class effectively initializes and displays the entire GUI.

- Sequential UI assembly
  - Updates splash progress and text via Jext static calls
  - Reads menu and toolbar definitions from XML resources
  - Registers plugins and executes startup scripts
- Workspace and file management
  - Loads saved workspaces
  - Opens files passed as constructor arguments in batch mode
- Dynamic pane configuration
  - Builds split panes for text editors, tool panels, and consoles
  - Toggles embedded tabs and content borders via client properties
- Interactive status update
  - Adds a mouse listener to rotate line endings on click
  - Updates "dirty" icons and status messages accordingly

Key instance fields and their roles:

- **defaultProjectMgmt**: Manages project operations and selection
- **inputHandler**: Processes user keyboard and mouse input
- **splittedTextArea, textAreasPane**: Represent dual text editor views
- **workspaces**: Loads and persists workspace configurations
- **textAreaSplitter, split, splitter**: Control JSplitPane layouts
- **vTabbedPane, hTabbedPane**: Hold virtual folders and consoles
- **console**: Provides an interactive command-line interface
- **leftFrame, rightFrame, consolesFrame, centerPane**: Frame containers

State mutations occur through:

- Static calls to Jext for properties, splash progress, and scripts
- Client properties on Swing components to alter appearance
- PluginHandler and ModeHandler listeners to react to runtime events
- WindowHandler to manage application exit and resource cleanup

Method	Purpose	Key Parameters	Return Value
`JextFrame(String[] args)`	Delegates to main constructor, showing GUI by default	`args`: array of file paths or null	New JextFrame instance
`JextFrame(String[] args, boolean toShow)`	Performs full GUI initialization and optionally shows window	`args`: startup file paths `toShow`: display flag	New JextFrame instance

Side Effects:

- Loads GUI components and plugins
- Opens files and workspaces
- Displays or hides the window based on toShow

- **Swing threading**: Initialization runs on caller thread instead of Swing EDT
- **Monolithic constructor**: Very long method hampers readability and testing
- **Static coupling**: Heavy reliance on `Jext` static methods reduces modularity
- **Hardcoded resource names**: XML filenames and property keys appear as literals
- **Exception handling**: No visible try/catch blocks; errors may be swallowed
- **Commented code**: Dead code fragments suggest incomplete cleanup
- **Thread-safety**: UI state mutations may conflict under concurrent events
- **Resource cleanup**: Lack of explicit disposal for frames and listeners may leak memory