The options presented by the export tool (likely a Jira plugin for requirements management) have a direct and significant impact on the quality of the data our script receives. Choosing the right settings will make the entire process more reliable.

Based on the goal of generating context-aware, granular test cases, here is the definitive recommendation.

Recommended Export Configuration

|  |  |  |
| --- | --- | --- |
| **Option** | **Recommended Setting** | **Reasoning (In order of importance)** |
| **Include folders hierarchy?** | **Yes** | **(CRITICAL)** This is the single most important option. It embeds the document's folder structure (ACC > System Logic > State Handling) into the REQIFZ file. This provides a rock-solid, unambiguous context for every requirement, which is far more reliable than having the AI guess the context from preceding headings. |
| **Export attachment previews?** | **No** | **(Performance)** The script is text-based; it cannot read or interpret images. Including previews will significantly increase the file size and slow down the parsing process for zero benefit to the automation. |
| **Export blank values?** | **No** | **(Cleanliness)** This creates a smaller, cleaner XML file. Our script is looking for specific, named attributes. Including dozens of empty tags for attributes we don't care about just adds noise and slightly increases parsing time. |

**Detailed Elaboration on Each Option**

Let's break down exactly *why* these are the best choices for our specific automation workflow.

1. **Include folders hierarchy? -> Choose YES**

This is the most strategic choice you can make for improving the automation.

* 1. **What it does:** When you check "Yes," the REQIFZ file will contain an extra section (<SPEC-HIERARCHY>) that explicitly defines the parent-child relationships between all your artifacts. It knows that requirement TFDCX32348-18153 lives inside the "Group 1" folder, which lives inside the "Basic Screen No.2" folder, etc.
  2. **Why it's CRITICAL for our script:**
     1. **It eliminates guesswork.** Our current script tries to guess the context by looking at the last "Heading" artifact it saw. This can be unreliable. With the hierarchy included, we can modify the script to *know* the exact path for every requirement.
     2. **It makes the AI smarter.** The prompt can be dramatically improved. Instead of just giving it a single heading, we can provide the full breadcrumb trail:
        1. **CONTEXT:** ADAS Display > ADAS\_ACC > Output Requirements > HMI Interface > Group 21: Preceding vehicle
        2. **PRIMARY REQUIREMENT TO TEST:** TFDCX32348-28189
  3. An AI given this precise context will generate far more accurate and relevant test cases than one that only sees the immediate heading.
  4. **Conclusion: Always export with the folder hierarchy. It provides invaluable structural metadata that makes your entire automation process more robust and intelligent.**

1. Export attachment previews? -> **Choose NO**

This is a simple choice for performance and efficiency.

* 1. **What it does:** If "Yes," it embeds image files (like PNGs or JPEGs of diagrams in your requirements) directly into the REQIFZ file.
  2. **Why it's BAD for our script:**
     1. **The AI can't see images.** Our entire workflow is based on parsing text. The AI models we are using (llama3, mistral) are language models; they cannot interpret pixels from an embedded image.
     2. **It creates huge files.** A single REQIFZ file could bloat from 1 MB to 50 MB just by including attachment previews. This makes unzipping and parsing the XML much slower, for no benefit.
  3. **Conclusion:** Choose **No** to keep the input file small, clean, and optimized for text-based processing.

1. Export blank values? -> **Choose NO**

This is a minor choice for file cleanliness.

* 1. **What it does:** If a requirement in JDA has 20 fields but only 5 is filled out, this option controls the output. "Yes" will export all 20 attribute tags, with 15 of them being empty. "No" will only export the 5 tags that actually have a value.
  2. **Why "No" is slightly better:**
     1. **Reduces Clutter:** It makes the resulting XML file smaller and easier for a human to read if you ever need to debug it.
     2. **Marginally Faster:** Less text for the XML parser to read through means a slightly faster script. The difference is negligible, but there's no reason to include unnecessary data.
  3. **Conclusion:** Choose **No** for a more compact and tidy input file.

By exporting with **Hierarchy: Yes**, **Attachments: No**, and **Blank Values: No**, you are creating the optimal input file for any advanced automation you want to build.

**Correct Options:**

