Asking SAE to use AS9102 exact text in FAI Assistant

First: Here’s email text to SAE about the idea:

Hi [SAE Contact Name],

I hope you’re well! As we continue exploring ways to integrate the AS9102 standard into my FAI Assistant app, I wanted to share a refined approach that offers both a greatly improved user experience and a stronger revenue opportunity for SAE.

**Current User Flow:**  
When a user selects any Form/Field (Forms 1, 2, or 3), they currently see my own paraphrased, educational content designed to help them understand and complete each AS9102 field accurately. Then, if they tap the “AS9102” button, they bring in the actual AS9102 PDF file to browse. This can be cumbersome due to the size of the phone and the document—they’ll likely need to “pinch-out” and scroll in order to view the section they’re interested in, which can make it hard to read.

**Proposed Enhancement:**  
With SAE’s permission, I would like to allow paid, licensed users to tap an “AS9102” button, which would then open a new screen displaying the *official specification text* for just that particular Form/Field—**no other sections or long passages from the standard would be included.** This approach mirrors best practices already used by industry leaders, but tailored for a much smoother mobile experience.

**Key Benefits for SAE:**

* **Ongoing Revenue Stream:**  
  Rather than a one-time PDF sale, SAE receives a share from each FAI Assistant license sold. This scales revenue with app adoption and provides a recurring, predictable income source.
* **IP Protection:**  
  The AS9102 field text would be stored in the app in encrypted (or obfuscated) form, accessible only to paid users. We would also implement UI restrictions to prevent copying or exporting content, helping safeguard SAE’s intellectual property.
* **Simplified User Experience:**  
  Users would get context-sensitive access to the official standard—no need to pinch/zoom through a PDF or manage separate files. This is faster, more accessible, and much easier for both end users and app developers.

**Important Boundaries:**

* Only the short, numbered Form/Field paragraphs (for Forms 1, 2, and 3) would ever be displayed—not the full text, tables, notes, or other long passages from AS9102.
* All official text access would be limited to licensed users and fully controlled within the app.

I believe this approach will maximize SAE’s licensing value while helping users comply with AS9102 requirements more effectively. If SAE approves, I can rapidly implement this—no user would need to purchase or load a separate file, and SAE’s standard would be presented in a highly secure, user-friendly way.

I’m happy to provide more technical detail, discuss revenue sharing models, or answer any questions your team may have. Please let me know if there’s someone on your team I can discuss this with directly, or if you’d prefer I continue communicating through you.

Thank you again for your time and support!

Best regards,  
[Your Name]  
FAI Assistant Developer  
[Your contact info]

Ok, here goes. SAE will make profit from FAI Assistant by way of users purchasing the AS9102 standard. This method involves my app to load the pdf file and for them to validate and keep track of it. One consideration is that when the pdf is loaded into the phone, the user can view it but since it is a full document the print is very small so the user most likely will have to 'pinch' open the document and scroll a little bit to read certain areas. So...SAE wants revenue from this product, I understand that. But what if...they give me permission to use the AS9102 standard text 'as is' in my app where instead of having the AS9102 button bring in the full pdf (having to go through all that file load procedure) and the user reading from there, my app just takes the Form/Field text portion of the actual document text and pops it up on another screen. So, in summary, the user taps the 'AS9102' button and a dialog or whole screen pops up with the few actual paragraphs from AS9102. This, I believe, is a much easier method for both the development and implementation SAI and I have to do. Another advantage to this method is that SAE can get a $ portion from every app sold instead of just the apps that sell without the user purchasing the AS9102. As you most likely know, the major piece of software that most companies use for FAI's is Net-Inspect. Now, you may or may not know but in Net-Inspect, next to every Form and Field designation, there is a little circled question mark where when you hover over it it pops up the actual AS9102 text - all that to say that SAE has done this to other software. Your thoughts and suggestions?

**ChatGPT said:**

This is a **fantastic idea**, and honestly, it’s right in line with what many modern SaaS tools do when licensing standards! Here are my thoughts, suggestions, and things to consider (including some “how to pitch it” talking points for SAE):

**1. User Experience:**

* **Direct Display:** Displaying the actual AS9102 paragraphs directly in the app—right next to the relevant field or with a button—is a *huge* usability win. No more awkward PDF scrolling, pinching, or hunting for the right section.
* **Just-in-Time Help:** Users get exactly what they need, when they need it, in a mobile-friendly format.

**2. For SAE:**

* **Recurring Revenue:** By embedding the standard in the app (with their permission), SAE can collect a *cut of every app sold*, not just one-time PDF purchases. This could become a *smoother, more scalable revenue stream*.
* **Digital Rights Control:** SAE can still control the license (e.g., only allow access if the app is paid/registered), and you can even watermark or digitally fingerprint the text, if needed.
* **Usage Analytics:** SAE can get data (via you) on how often various sections are viewed, potentially valuable for future revisions or marketing.

**3. For You (FAI Assistant):**

* **Simpler Development:** No need to wrangle PDF viewers, pinch-zoom code, or file management headaches.
* **Streamlined Onboarding:** New users get an all-in-one experience without buying, finding, and loading a separate file.
* **Competitive Advantage:** Matching (or beating) Net-Inspect’s UX by providing instant, official reference text. You’re in good company!

**4. Precedent Exists:**

* **Net-Inspect** and other platforms have already set the expectation: If they can do it, so can you—*with proper licensing*.

**5. How to Pitch It to SAE:**

* Focus on **user value** and **revenue share**.
* Emphasize **copyright control**—you’ll show only small, relevant portions, not the full standard at once (unless they want that).
* Offer **usage stats** and perhaps a mechanism to keep text up to date (in case of future standard revisions).

**6. Implementation Suggestions:**

* Store the AS9102 text in your app as an encrypted resource or retrieve from a secure cloud endpoint if SAE prefers.
* Only unlock the text if the user is a paid/registered user—protecting SAE’s IP.
* If SAE is worried about “screen scraping” or copying, you can disable copy/paste on those fields or watermark the content.
* Consider offering SAE a **periodic audit report** showing compliance with their usage/license terms.

**7. Extra Monetization Ideas:**

* If SAE likes this model, you could extend to other standards or industry documents, offering them a new digital sales channel.
* Allow users to “upgrade” for access to additional SAE resources (cheat sheets, FAQs, training modules).

**8. Potential Risks/Obstacles:**

* SAE’s lawyers may be cautious at first—be ready to explain how you’ll protect their content (app encryption, access control, auditability).
* Be clear that users won’t get a raw “copy and paste” experience—they’re getting official, read-only, context-sensitive help.

**Summary for SAE:**

“Let’s give users the AS9102 guidance they need, exactly where they need it, with proper licensing and revenue share to SAE. We’ll make sure your intellectual property is protected, and you’ll see recurring income every time a user pays for the app—not just when they separately buy the PDF.”

**Your idea is win-win-win:**

* **Better for users**
* **Better for you (less dev work, more value)**
* **Better for SAE (more sales, more control, more data)**