PUBLISHING YOUR APP

Minimizing the mockery

A person who publishes [an app] appears wilfully in the public eye with his pants down.
—Approximately Edna St. Vincent Millay

Now that you've learned how to write ServiceNow code that is only marginally embarrassing[1], you may find yourself wanting to publish your application on the ServiceNow store. If you've never published an application to the ServiceNow store, you may be wondering "how difficult could this possibly be?"

Oh, you poor fool. How I envy your ignorance. There are many rules and risks to keep in mind at every phase; from planning, to building, to publishing and re-publishing. This *sphincter-clenching, balls-out[2], action-packed chapter* will prepare you for some of these pitfalls, things you can do to set yourself up for success, and some bonus tools and techniques to build your app for a great customer/user experience.

◆ ◆ ◆

---

[1] *Threshold of marginal embarrassment may be exceeded. Humiliation, Mockery, and "Loser" are trademarks of Tim Woodruff, Inc. For detailed side-effects of public humiliation resulting from publishing your application, consult with your doctor. If you are not subject to the American healthcare system or television ads, you may not get this joke. Taking this joke in conjunction with low Freedom™ levels may cause confusion, dizziness, and early-onset diarrhea.*

[2] *The phrase "balls out" refers to the centrifugal governor in an old mechanical steam engine. When running at full speed, the brass balls in the governor move outward due to centrifugal force, limiting the steam engine to avoid running out of control.*

PLANNING

Once your app is submitted to the Partner Portal for review prior to publishing, it'll be inspected for security, privacy, performance, stability, and user experience. In addition to the app itself, you will need to provide a complete listing for the app as well artifacts speaking to its Design, Test Plan, and Installation Guide.

The planning phase of any application you intend to publish is particularly important and only becomes more so when you begin building version 2 and onward.

Planning for the future before you begin building is **always** important, but planning for v2 and onward before you publish v1 is *even more important* when it comes to applications on the ServiceNow store. Some specific examples of why this is, include the fact that deletions are not deployed with future version updates of your applications. You might think that's not a big deal, but consider how many actions are technically deletions…

Consider ACLs, for example. Imagine you'd like to remove a role from an ACL in V2 of your application. ACLs live in the `sys_security_acl` table, but the roles associated with ACLs live in a separate M2M (many-to-many) table: `sys_security_acl_role`. If you want to remove a role from an ACL, you have to delete a record in the `sys_security_acl_role` table. That deletion cannot be contained within a store app after it's initially published.

Fun fact: If you *do* want to remove a role from an ACL in your app after it's been published, the "cOrReCt" way to accomplish that is to go to the ACL *Role* record and *clear the reference field that associated it with the ACL…*

…Seriously. That is the actual "proper" way to remove a role from an ACL in a published ServiceNow store application.

Hopefully this elucidates the importance of planning your application's future as much as possible, in the early stages.

GENERAL REQUIREMENTS

So, you have an absolute banger of an idea and want the world to know about it? Maybe you'd like to make some money on your great idea for an AI-powered app that puts Web3D-prints of your NFTs in the Metaverse's blockchain? This chapter will help guide you on how to build and publish that terrible idea for all the world to see and laugh at!

Not just anyone can apply for their app to be certified on the ServiceNow Store (https://hbk-sn-store.snc.guru). You (or the company you work for/with) must be part of ServiceNow's Build[3] program. At the time of writing, it costs ~$5000 USD to register. This fee gets you into the "TPP" (Technology Partner Program) and grants you a "Vendor Instance" (typically with a subdomain starting with "ven", like "ven012345.service-now.com"). This instance will have your "vendor application scope prefix" on it, and it is the instance on which you *must* build your ServiceNow application if you want to publish it on the ServiceNow store.

__Pro-tip:__ Technically, you __could__ still publish your application on the Store if you built it on another instance like a PDI. You can do so by following these steps:

1. *Create a new empty app on your TPP "Vendor" instance to create a new app scope. Copy that app scope prefix. It'll be something like* `x_myCompany_123`.
2. *"Publish" the app you built on your non-vendor instance to an Update Set.*
3. *Export that Update Set to an XML file.*
4. *Open that Update Set XML file in a text editor such as Notepad++.*

---

[3] *Become a Build Partner: https://hbk-build-partner.snc.guru*

5.   *Do a find-and-replace (usually CTRL+H), and replace any instances of your old application scope prefix with the new vendor app scope prefix.*

6.   *Save the Update Set XML file.*

7.   *Import your application's updated Update Set into your Vendor instance, then preview and commit it.*

You should now be able to run the SN Store self-test tool and publish your application. However, keep in mind that I can't promise that this will continue to work, and I'm not responsible for any time you waste in your PDI if it doesn't!

If you're an independent contractor wanting to have your app certified and published, you can pay the (imho ridiculous) fee and register yourself into the Partner program, or you can reach out to already-certified ServiceNow partners for a collaboration, but unfortunately you cannot simply build a ServiceNow "Store app" on a PDI (Personal Development Instance) and publish it on the Store directly.

More information about requirements and detailed descriptions of each step can be found at the Technology Partners Portal[4]. However, only registered Build Partners have access to the partner portal site and all of the knowledge articles and documentation that can be found there. ಠ_ಠ

Once you are registered as a Build Partner, you will be able to create and build your app within the provided Vendor instance. Once you have completed development of the app, ensure that the Certification Self-Test Tool[5] plugin is installed on the instance and is upgraded to the latest version. This tool will run automated checks on your app that comprise a lot of the most common issues that prevent application certification. Running the certification self-test tool on your application in your vendor instance and either resolving or documenting *responses* to any findings it produces, is **required** before you can submit your application for review and publication. Don't worry, the partner portal submission form knows whether you've run the self-test tool on the latest published version of your app or not. You won't be sitting around waiting days for a certification tech just to bounce it back to you, saying "run the certification self-test tool and try again".

Want to add `current.setWorkflow(false)` in your "before" Business-Rule? Not in this house, no sir! The Self-Test Tool will highlight this – and some (but certainly not all) other common application certification issues – for you to remediate prior to beginning the certification process. You may also – if you are an astute and erudite observer of technology - that not all of ServiceNow's base-platform code is run through this tool, or the often very thorough manual ~~colonoscopy~~ review that follows. You may note, for example, that certain OOB (out-of-box) code is highly non-performant and not even remotely in alignment with ServiceNow's own stated Best Practices[6]. Well… while we all surely hope that ServiceNow employs the same rigor to its own code in the future, let's at least ensure that ours is the best it can be.

Some common reasons for failing certification checks are missing roles for application navigator modules, dot-walking to the `sys_id` field (resulting in an unnecessary union, impacting performance significantly, as described in detail in the Performance chapter of this handbook), missing ACLs for UI Pages or client-callable Script Includes, and many more[7]. Outside of the Self-Test Tool, there are plenty of other common issues to avoid or consider. Ensure your app does not extend Incident, Problem, Change, or Catalog task tables. Avoid altering of create/read/update/delete access to OOB (or plugin-related) tables. Things like that.

There are additional requirements that pertain more to application-specific development (e.g. Vulnerability Management, CMDB Integrations, Service Portal, etc.). These requirements are mostly around confirming the OOB functionality is used as part of the app wherever possible. To put it succinctly - a custom app cannot replace OOB functionality to circumvent licensing one of ServiceNow's own apps.

---

[4] *Partner program portal: https://hbk-tpp.snc.guru*

[5] *Certification self-test plugin installation and usage:*
https://hbk-self-test-tool.snc.guru

[6] *https://gr.snc.guru & https://hbk-single-record-violations-gist.snc.guru* ಠ_ಠ

[7] *Community article for the top 10 most common reasons an app fails certification: https://hbk-common-fails.snc.guru*

SECURITY REQUIREMENTS

There are several Security-related requirements that your app must adhere to, in order to be published on the ServiceNow store. The first is around authentication and authorization: custom authentication methods are not permitted on the ServiceNow Store. You also have to restrict roles to only the necessary permissions required (e.g. integration roles should not have 'admin' or ITIL or anything else that seems unrelated). And prepare for some questions from your reviewer if you incorporate roles from other app scopes into your own app!

The permissions requirements extend to the rest of the application as well. ACLs and related Roles should be created to protect Tables, UI pages, Widgets, and other components. When setting up these Roles, ensure there is at least one for your app and that you follow the least-privilege principal, whereby the least amount of access is given to satisfy the access requirements (including for cross-scope privileges).

Data protection is also a must-have for your application. Ensure encryption standards are followed (AES, not 3DES) and any sensitive data is encrypted with system properties (password2) or `encrypted_text` fields. Additionally, using `GlideEncrypter` and `ScopedEncrypter` are not permitted as part of a scoped application, published on the app store... Despite the name of the latter.

Be doubly sure that logging of sensitive data is prevented. This includes taking steps to do things like prevent credentials from being stored as part of a target URL for messages sent via the `ecc_queue` table, but it also includes being careful not to throw errors or otherwise log messages that could potentially include sensitive data. Of course, as mentioned in the **Performance** > **Unnecessary log statements** section, always make sure not to log unnecessarily _at all_! But do remember that even in your `gs.debug()`, `gs.warn()`, and `gs.error()` messages which may need to remain part of your app's code, you should never log anything that may contain sensitive information. That includes complex objects or arrays which may contain a credential value, CUI (Controlled Unclassified Information), or PII (Personally Identifiable Information).

If you need to register external users as part of your app, you'll need to use the External User Registration plugin[8] (`com.snc.external_user_registration`) in conjunction with the externally facing portal pages the users will access.

PRIVACY REQUIREMENTS

ServiceNow requires that your application has a comprehensive Privacy policy, which details how data (e.g. personal data, user records, all record types, etc.) are collected, stored, and shared. This policy must be within the app listing on the store, and any updates to it must include some kind of notification to your users. Any personal data that is collected from the user and sent outside of ServiceNow must be preceded by an explicit consent form, informing the user of its collection and processing. Users must also be able to access, correct, and delete their personal data.

As part of the Privacy policy, you'll need to state the retention period for personal data and any criteria used to determine the specified duration. Your app will then need to anonymize or delete the data after the defined retention period. These measures help to ensure the app will adhere to applicable data protection standards, but there may be additional laws and regulations within your user's countries or localities which you'll need to take into account. Europe's GDPR and California's CCPA are a good starting point to ensure your app covers privacy regulations, but you may need to familiarize yourself with other countries as well (e.g. Australia's Privacy Act, South Korea's Personal Information Protection Act, etc.).

---

[8] _Fun fact: You can add other apps as a dependency of your app, by going to your app's **Application Settings** page from ServiceNow's Studio UI, navigating to the **Dependencies** related list, and adding a new dependency. You can even optionally add a minimum version number to the dependency registration._

PERFORMANCE REQUIREMENTS

The next lens through which your app will be analyzed by ServiceNow, is Performance. If your app queries large tables like the log or audit tables, you'll likely want to ensure indexes are in place for your common "where" clauses (query filters) and sorting. Similarly, if your app requires data transfer over the internet, aim for minimal payload size and frequency to prevent straining the platform causing performance issues elsewhere. If it looks like your app may significantly impact the performance of the customer's instance, there's a good chance it won't be allowed on the app store.

Large scripts should also be refactored into reusable Script Includes wherever possible. You should make an effort to modularize your code for better maintainability. For client-side calls to server-side code, `GlideAjax` should be used with a callback function. (More on GlideAjax, how to use it, and why it's important, in my article on SN Pro Tips, at https://ajax.snc.guru.)

GENERATIVE AI REQUIREMENTS

Getting in on the next hot thing so you can get paid before everyone realizes it's a bubble? You'll need to ensure your app has a disclaimer for any GenAI functionality, which states: "*ServiceNow is not responsible for the accuracy, completeness, legality, or appropriateness of any content generated with Generative AI.*"[9]

You'll need an additional disclaimer[10] to the user in any interface where you present AI-generated ~~slop~~ content, that states "*This content is generated using Artificial Intelligence (AI). Please review it thoroughly for accuracy and relevance.*" You will also need to indicate the source of the generated content within the app (e.g. Gemini, ChatGPT, or ServiceNow's GenAI).

The AI models you select for the app are also important. Ensure that the models you use are not deprecated or unmaintained, and that the models meet established security and privacy standards – including keeping things up to date as those standards change over time. Further, the model(s) you select will need to comply with community standards that avoid harmful or inappropriate content.

LICENSING

NOTE: In this brief section, I'm going to talk about licensing. – I know. Bold, right?

Listen. I am not a representative of ServiceNow. I'm just some dumb guy. I don't know what I'm talking about, okay? Don't rely on what some dumb guy wrote – no matter how charming and attractive he may be – as your sole source for something as important, as convoluted beyond comprehension, and as constantly changing as ServiceNow's licensing schemes. I'm basing what I know on input from the community, and on public documentation – provided by ServiceNow and referenced where appropriate – but things could change, and I could be wrong.

Okay. We good? Okay. Here we go.

Typically, if you're **charging a license fee for your application** on the ServiceNow store, your customers don't *typically* have to pay for certain things contained within your application. For example, if a customer has a limited number of custom tables

---

[9] *It's always a good omen when a technology comes with a ubiquitous disclaimer telling you how confidently wrong is constantly and unavoidably is.*

[10] *As GenAI is relatively new, these ServiceNow-provided disclaimers may be updated. The content of the disclaimer should therefore be somewhat modular (maybe referencing a system property to store the disclaimer message?) so you can bring in the new messaging quickly.*

they can have in their instance before triggering licensing implications, and they buy your application with ten additional tables, then those ten tables should *not* count toward their limit or result in additional fees from ServiceNow.

However, if your application is **free**, then this no longer applies… your customers *may indeed* be charged for any custom tables in your application, if it is free.
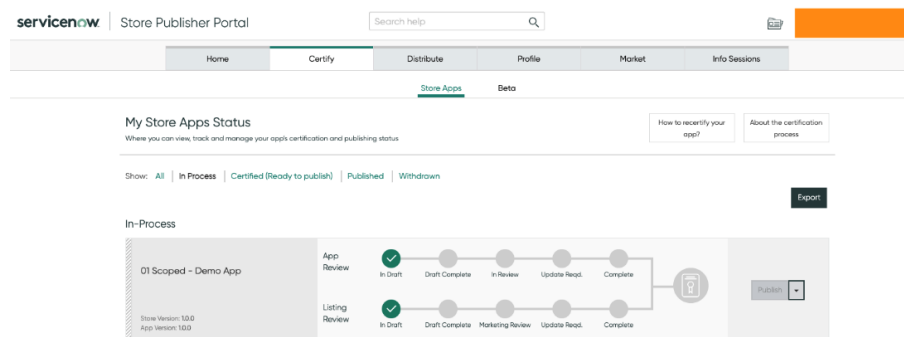
There is, though, a way around that – at least, last I checked. If your application type is set to "**Integration**", then those limitations **do not apply**[11].

There are many caveats when it comes to ServiceNow's licensing model, but the all-important, ever-present, and always accurate advice still holds true: the most important thing when trying to figure out licensing implications in ServiceNow, is to (1) contact your account or partner representative, and (2) get their answer documented in writing over email.

PUBLISHING YOUR APP

This section is applicable only after the app has been reviewed for the above criteria and the Certification Self-Test Tool has been run within the Vendor instance[12]. Your app must be published before you're able to draft your app listing for the store, submit your documentation for review, and begin the certification review process. Once you have completed your internal reviews and run the Certification Self-Help Tool, the app may be submitted for the certification process to begin. This submission can be done within the Vendor instance, from the Studio… or the "ServiceNow Studio". Either one.

You can also submit the application directly from the app record: In your Vendor instance, open the left-hand nav and go to System Applications > My Company Applications, select the app in the list. On the application record form scroll down to the **Related Links** section and click on Publish to ServiceNow Store[13]. At this point, there will be a record in the Store Publisher Portal for your app and app listing in the state of In Draft, shown below.

You'll need to open up that draft application record, fill out all required information and provide the required artefacts mentioned above. Once that's done, you can submit your application for review, then cross your fingers and wait.

---

[11] *This is according to the ServiceNow Custom Table Licensing Guide, of which you can see the latest available version as of the time of writing (last published April 2025), here:*
https://hbk-custom-table-licensing.snc.guru
[12] *Lest you want your app immediately sent back to fix issues mentioned in the sections above.*
[13] *Guide for uploading an app to the store for review (steps 1 thru 3 of the doc):*
https://hbk-app-upload-guide.snc.guru

APP LISTING REQUIREMENTS

Your application will require a "listing" page for the ServiceNow Store, which you'll be able to create once you've submitted your application. This listing is a separate set of information and artefacts from the app info, and has a separate (but typically much faster and simpler) submission, review, and approval process. The main sections needed for the listing are details about the app called "App details" and "Version details". The app listing cannot be worked on until the app is submitted to the Store[14].

Under "App details", you'll begin with the summary which includes a brief description of the app and accompanying screenshots or recorded demos[15]. Below the summary is the description and key features of your app which allows for more exposition of the app's functionality along with a bulleted list of its most appealing features to your target audience[16]. You'll also need to include release notes for each version of the app after the first. These can be used to highlight new features or changes to the app from previous versions.

A screenshot of the app listing form on the ServiceNow Partner Portal

Other information your app listing will require you to include a description of the requirements for your app to run correctly. These requirements take the form of any external subscriptions needed for the app, external access or license requirements (does the user need an account in another system, or to purchase other software which this app integrates with?), and any ServiceNow plugin dependencies (*especially* if those plugins come with a potential license cost).

---

[14] *For creation and submission of the app listing, steps 1 through 4 of this article must be completed: https://hbk-app-creation-submission-guide.snc.guru*

[15] *It's not a bad idea to click through other apps on the ServiceNow Store to get a feel for which screenshots and marketing materials might convey the importance of your app most clearly.*
https://store.servicenow.com

[16] *For more info on marketing materials, go to the Partner Success Center here:*
https://hbk-partner-marketing-materials.snc.guru
Note: you must be registered as a Build Partner to have access.

REQUIRED LISTING ARTIFACTS

To complete the draft stage of your application, you'll need to upload accompanying documentation in the Store Publisher Portal. There are two documents that are only for the certification team and one that is customer-facing (for your "listing").
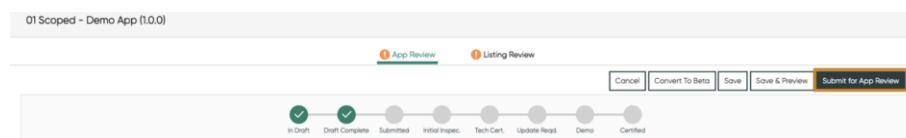
The first document required for review by the certification team is for the **design** of your app[17]. You will need to create a Design Document, starting with a template available on the ServiceNow Store website. This document includes your business case, outcomes, features, constraints, lifecycle of the app, and how the app will be supported in the future. The design document also requires a thorough explanation of the Solution Architecture which is comprised of the basic components of the app, descriptions of the tables and data involved, business logic, integrations, authentication, access management, and performance. The document may be bolstered with images that further explain your app, such as business process flows, use case models, swim lane diagrams, and entity relationship diagrams.

The second document the certification team will need to approve your app is your **testing plan**. This will need to cover how the app is tested and examples for ATF (Automated Test Framework) if it is being used. A template for this document is also available on the ServiceNow Store website.

The last document that is needed is the Installation Guide. The installation guide covers an overview of the app's functionality, any dependencies of the app, instructions for configuration, integrations, testing steps, a description of the included demo data, and steps for troubleshooting the app. As this is a public-facing document for each application, read through a few that are attached to other store listings for examples of what is needed. It's a good idea to have a Guided Setup in your application for many reasons, but being able to just point to that info in your installation guide can be a real time-saver.

CERTIFICATION PROCESS

At this point in the certification process, you've run through the manual checks (Security, Performance, etc), run the Certification Self-test Tool, uploaded supporting documentation (Design, Test, and Installation Guide), and completed the draft of your app's listing on the Store. Under the App Review section in the Certify tab of the Publisher Portal, click on the "Submit for App Review" button in the upper-right. This button begins the process of reviewing your app for the technical standards mentioned in the sections above, including an often-very-thorough manual review by a ServiceNow employee. A similar but much more lightweight process happens when you submit your app for "listing review" under your app's Listing Review tab in the Publisher Portal.
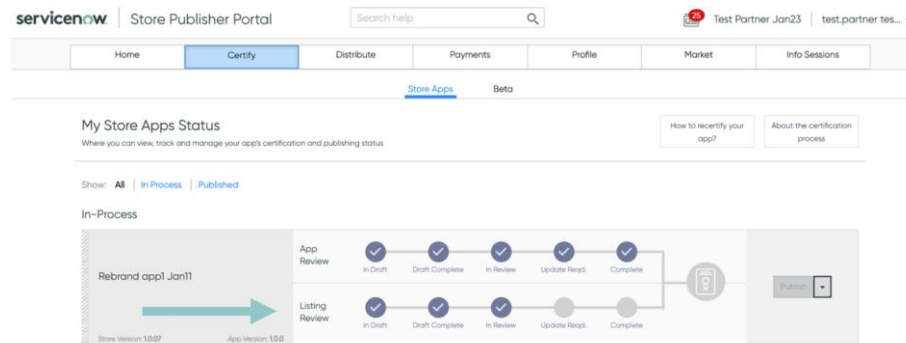

Once all feedback for the app and the listing have been incorporated, resubmit both sections of the form for final review. When both certification workflows have been completed, you may finally click the **Publish** button, shown below to the right of the workflows; though it will no longer be greyed out once your app is ready to be published.

Click Publish to make your app available for customers on the ServiceNow Store, and that's it. You are now a published app author!

---

[17] *This document gives templates for the Design doc, Testing Plan doc, and the Installation Guide:* *https://tpp.servicenow.com/sn_appstore_store.do#!/tpp/program?article=KB0030070*

SUMMARY

ServiceNow's Store is similar to the Apple or Google app stores in that there are barriers to entry to ensure each app meets certain standards and governmental regulations. To ensure that these standards are met, ServiceNow reviews apps for Security, Privacy, Performance, Stability, and User Experience. The Store listing further ensures that each app lists plainly what its purpose is, how much it costs, what it depends on, how to install it, and what the product will look like when it's in-use. After the technical reviews are completed and the marketing (app listing) reviews are completed, you too can have your very own app on the ServiceNow Store.

Here's a summary of the typical app certification process:

- Ensure your company is a registered Build Partner
- Obtain a Vendor Instance or access
- Create the app in the vendor instance
- Run through the Security, Privacy, Performance, and GenAI requirements (if applicable) as mentioned in the relevant sections in this chapter
- Run the Certification Self-Test Tool
- Publish the app to the store
- Draft the app listing and attach the required documentation (Design, Test, and Installation Guide)
- Submit the app and app listing for certification to begin the formal review process
- Fix any (hopefully minor) feedback from the Store engineers
- Re-submit the app and app listing for certification if necessary
- Publish your app and listing on the ServiceNow Store
- ???

PROFIT!