**XLAutoDeploy Configuration Notes and Ideas**

Application Manifest: <https://docs.microsoft.com/en-us/visualstudio/deployment/clickonce-application-manifest?view=vs-2019>

* Specific to a single version of a deployment; b/c of this, they should be stored separately from deployment manifests
* Copy to client machine: Common convention is to place them in a subdirectory named after the associated version.
* Stores URI of Deployment manifest
* This should be an xml file and be signed.
* Naming Convention: applicationName.exe.manifest

Option to specify application’s required truest level

Deployment Manifest: <https://docs.microsoft.com/en-us/visualstudio/deployment/clickonce-deployment-manifest?view=vs-2019>

* Resides on server only

points to the current application manifest

Update Strategies: Run check on start up

<https://docs.microsoft.com/en-us/visualstudio/deployment/choosing-a-clickonce-update-strategy?view=vs-2019>

* Silent updates: Cannot install the application
* Installs: Must restart excel application. This would be require a UI to notify the user.
* Updates can be required (developer would need to specify a minimum version) or optional.
* Applications that require per machine installs must run under elevated permissions. These permissions would have to be granted by an admin.
* Subsequent versions of the application could require elevated permissions, so need to a have a way to specify application’s required permissions.
* Update based on time intervals.
* Update based on user trigger event or some other event.
* Ability to specify per-machine (requires admin permissions) or user-profile installs.
* The first time an add-in is loaded/installed on the machine, the user should be asked if they trust it. Every time there after (unless required permission levels have changed) the user should NOT receive a prompt. This notify should still be optional.

Only Download files if they have changed.

* Check changes by comparing hash signatures of file specified in the application manifest (on the client machine) to the incoming (application or deployment ?) manifest

Deployment

Publishing:

<https://docs.microsoft.com/en-us/visualstudio/deployment/publishing-clickonce-applications?view=vs-2019>

Licensing notes:

1. Client generates a unique hardware ID based on information gathered from the machine (MAC, motherboard/hdd/cpu serial numbers, etc)

2. Client sends this unique ID along with some kind of license identifier (a key) to a remote system (either via internet or any other mean)

3. The server looks up the license from the identifier, and prepares a payload containing information such as the expiry date, and other details of the license such as who owns it and any applicable restrictions - the payload is then encrypted using reversible encryption where the unique hardware ID is the encryption key.

4. The encrypted payload is cryptographically signed and sent to the client. This way the client can verify it came from you.

5. The client verifies the signature of the payload against a known public key, and decrypts the payload using its unique hardware ID. If either of these verifications fail then the license has either been moved or tampered with.

In the above model, for increased security, each client should have their own “public key” that only has access to secured remote resource scoped only to that client. If a single public key allowed access to all remote resources, then your entire repository would be compromised if someone were to discover that key.

A trusted organization assigns a certificate to an individual or an entity that associates a public key with the individual.

Target a Windows Client

Depending on windows 10 versions, there are different default .Net Framework versions installed as the OS component.

https://docs.microsoft.com/en-us/archive/blogs/astebner/mailbag-what-version-of-the-net-framework-is-included-in-what-version-of-the-os

https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies

* The manifest files should have the ability to package runtimes (i.e. prerequisites) for install.
* What kind of user permission(s) (if any) are required to install? Admin
  + Need to deploy a separate program to install, or have users, or organization install it themselves.
  + What about registering unmanaged assemblies?
  + Excel xlam, or xla files that depend on dlls? These dlls would have to be signed using the following method: https://stackoverflow.com/questions/37074533/how-to-call-net-methods-from-excel-vba, <https://stackoverflow.com/questions/19954001/calling-a-net-library-method-from-vba>

<https://stackoverflow.com/questions/64035439/excel-dna-addins-how-pack-unmanaged-dll-in-xll-file>

<https://stackoverflow.com/questions/58139068/packed-excel-dna-xll-file-does-not-load-ribbon>

<https://stackoverflow.com/questions/52735817/excel-dna-add-in-with-connection-strings-stored-in-external-configuration-file>

Document Signing:

<https://docs.microsoft.com/en-us/dotnet/standard/security/how-to-sign-xml-documents-with-digital-signatures>

<https://stackoverflow.com/a/359923>

<https://docs.microsoft.com/en-us/dotnet/standard/security/cryptographic-signatures?redirectedfrom=MSDN>

<https://docs.microsoft.com/en-us/windows/win32/seccrypto/cryptography-tools>

Check local files to see if they match the key from the deploypment manifest.

what if a file is intercepted by a hacker and modified before being passed through the download?

If hashes are pre-computed and stored in the application manifest, which is deployed to the client, then upon deplopment of updates, there file

add-ins can always be re-loaded silently on startup if they are never installed, thus you would never have to restart excel.

append "legal" file extentions and remove at runtime

prevent multiple installs?

%PROGRAMFILES% is not writable by regular users from XP on.

RollBack Update????

System.Deployment.Application: <https://docs.microsoft.com/en-us/dotnet/api/system.deployment.application?view=netframework-4.8>

LocalizationSupport: A satellite assembly is an assembly that contains application resources for a culture other than the application's default culture.

If addin is Com, then use interop to connect/disconnect on startup: <https://stackoverflow.com/questions/48565694/enable-com-addins-in-excel-through-vba>

VSTO Add-in Architecture: <https://docs.microsoft.com/en-us/visualstudio/vsto/architecture-of-vsto-add-ins?view=vs-2019>