Winston & Strawn - Asset Tag App

Client	Freshworks
Product	Freshservice
Customer/Company	Winston & Strawn
Client Contact(s)	Uday Grewal
Client Contact Email	uday.grewal@freshworks.com
Author	Prasanna Balasubramaniam
Version	1.0
Project Start Date	TBD
Reference IDs	FW 1109

INTRODUCTION

To create a custom app for Winston & Strawn's Freshservice instance that generates sequential asset tags and updates asset records in Freshservice accordingly.

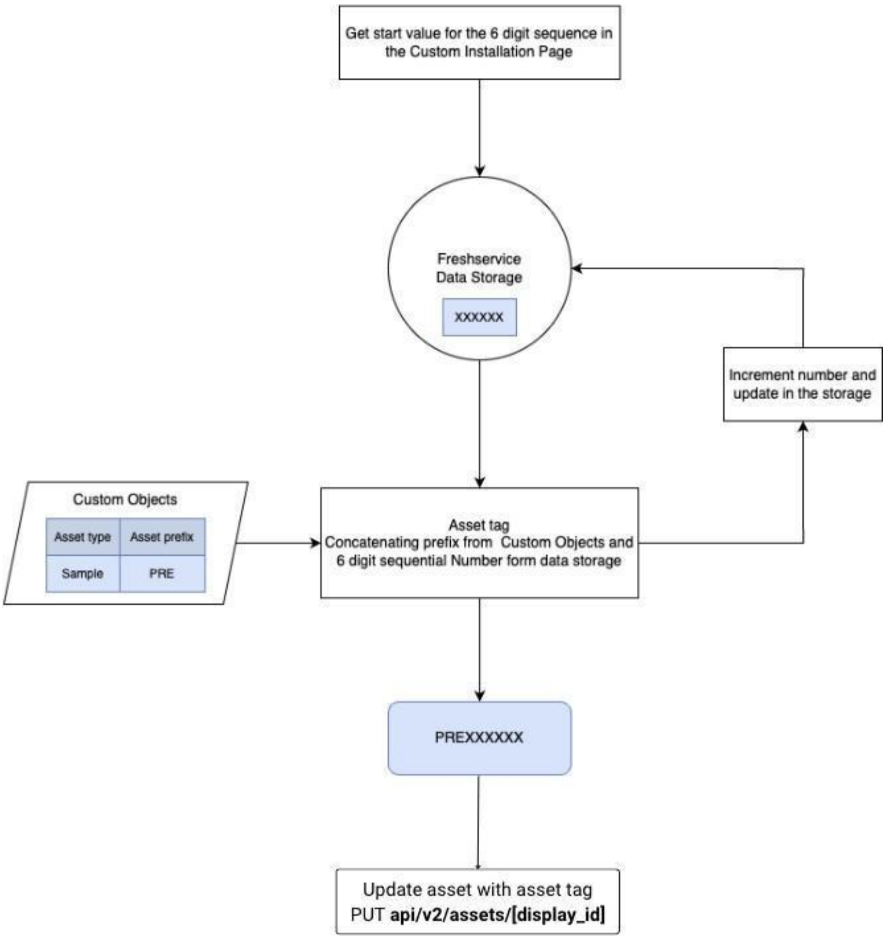
SCOPE OF WORK

The following are the terms of this engagement:

- The app has to sequentially generate an asset tag via workflow automation (9 characters long) and set it to the asset when it's created. The prefix (first 3 characters) will be a value associated with an asset type. The succeeding 6 characters will be a 6-digit number that increments by 1 for each asset created.
- Develop, deploy, test, and UAT

DEVELOPMENT IMPLEMENTATION

The following describes the high-level architecture for handling the asset tag by taking the 6-digit sequence entered on the app configuration page and storing it in the Freshservice data storage.


Get start value for the 6 digit sequence in the Custom Installation Page
Freshservice Data Storage
XXXXXX
Increment number and update in the storage
Custom Objects
Asset tag
Concatenating prefix from Custom Objects and 6 digit sequential Number form data storage
PREXXXXXX

Asset type	Asset prefix
Sample	PRE

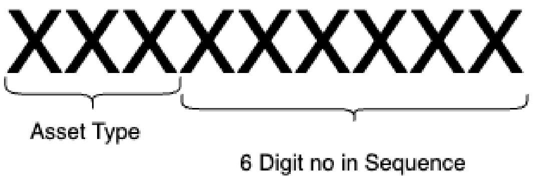
Update asset with asset tag
PUT **api/v2/assets/[display_id]**

Asset Tag: Auto Increment

1. A configuration page will be available to specify:
    a. API Connection Parameters (Freshservice Domain and API Key)
    b. Starting number for the 6-digit code (this will be available for configuration right after installation and will subsequently be disabled)
2. Workflow automation will be configured to trigger a webhook when an asset is created.
3. This webhook will be registered as an external event in the app.
4. When called, the asset will be updated using the assets API.
5. The prefix will be fetched from Custom Objects based on the asset type selected.

6. The six-digit tag number will be stored in the Freshservice data storage. The number will be incremental for each asset, despite the asset type.
7. After concatenating the prefix and the six-digit number, the asset will be updated with the resulting asset tag via the Update Asset API call
8. The old tag number in the data storage will be replaced with the new value.

Format of the Asset Tag:



Custom Objects:

- By creating a custom object with asset type and prefix as key-value pairs. The app will search the custom object records (via the Custom Objects API) matching the asset type to extract the prefix.
- And then, the prefix will be concatenated with the 6 digit sequential number.

Example:

Following are some of the Asset Types from which the prefix will be extracted and concatenated with the sequential number. This formation is pushed into the Asset Tag field (column). "CO" refers to Custom Objects.

Asset Type in CO	Prefix in CO	Sequential Number	Asset Tag
VMware VCenter Datacenter	VMW	202303	VMW202303
AWS	AWS	202304	AWS202304

GENERAL ASSUMPTIONS/QUESTIONS

1. The sequential format of the 6-digit character should be defined by the customer. Example: From which number to start so that the sequence can be followed.

2. The custom object records and mappings (3-character prefix code for each asset type) will be created and managed by the customer (and not by Effy). The customer will create and maintain mappings for asset types.
3. If a mapping does not exist, the app will not set the asset tag for the asset.
4. The asset tag will be set only during the creation of the asset during the asset lifecycle.
5. In the event of validation failures while updating the asset via the API, the asset tag will not be populated.

DELIVERABLES

The deliverable is:

- Freshservice Custom App: The app will be developed to generate sequential asset tags and update asset records accordingly.

SPECIFIC EXPECTATIONS

The following specific expectations are necessary to build the integrations:

Freshservice Sandbox Account	A Freshservice account will be required during development and internal testing.
Custom Objects	Custom objects should be pre-created and available prior to the development.

ASSUMPTIONS

API/Platform Limits on Freshservice

The FDK Request Method has set limits and criteria for how the Freshdesk API should work in the marketplace. This includes the response timeout, request, and response adherence in the documentation. Freshdesk will define sufficient product APIs and platform limits for the integration to work.

Changes to Scope

Any changes to the scope will be mutually agreed upon, and the stakeholders will sign off on the revised Statement of Work document.