

Note: This document is a writing sample for Joseph Perez which has been anonymized/fictionalized. Code referenced herein is also fictionalized or omitted. The original doc was written in Markdown.

# What Are Starnotes?

*Use Starnotes to add textual and visual enhancements to Spacecharts*

## Contents

### Introduction

- The many uses of Starnotes
- Starnotes and crew manifests

### Key elements

- Starnote protos
- USE\_CASE
- Transform
- SubnoteContent
- Target

### Producing and serving Starnotes

## Introduction

Starnotes are a set of APIs for altering Spacechart visual content with semantic notes such as a text, icon, or hologram. There are two major systems involved in putting data on the Spacechart as a Starnote: a set of two backend protos that integrate with the Spacechart serving stack and a single set of frontend protos used across the main Spacechart client surfaces to modify chart creation or behavior based on those Starnotes.

The Starnotes can live inside the space-units produced and served by SpaceMaker, but they may also be produced by other Rokket backends. Once a Starnote is produced, it is sent to and drawn on the client devices used by rocket crews. Even if an older client does not support Starnotes, they can still be added to the chart surface by Rokket before they are served to clients.

There are many different appearances of StarNotes on Spacecharts:

[SCREENSHOTS OMITTED]

Top to bottom: Starnote hologram added to Sol solar system chart, Starnote text captions added to Sagittarius, Starnote data added to rocket ship's bridge viewscreen.

## The many uses of Starnotes

Starnotes are a good fit for Mission Control feature owners who want to add a small bit of information to crew manifests and other data on the Spacechart. They are an especially good fit for various conditions:

- for content that isn't meant for all chart views (such as asteroids or moons in navigation)
- for drawing information that can only be determined on the client device (e.g., if a crew member has already interacted with a chart query in the current session)

Starnotes must target an existing manifest. The manifest can be:

- part of space-units (e.g.: a feature that you already see in a collection of space-units on the Spacechart)
- client-added (e.g., a rocket crew's portable tricorder)
- boosted from a manifest that exists from long distances tracked by the FarAwayMeter.

NOTE: Although Starnotes can theoretically apply to any Spacechart feature, they're only implemented for manifests at this time. If you need to alter a non-manifest Spacechart feature (e.g.: dark matter), reach out to the [Space Chart Team](#).

Starnotes can alter the existing manifest in the following ways:

- add Subnotes to provide non-essential but useful details
- change drawing method (e.g.: hologram transparency or text color)
- have viewport-specific or crew-specific parameters (these modifications will require further backend integration)
- increasing or reducing the FarAwayMeter levels at which it appears

★ TIP: While you're considering a feature that uses Starnotes, please contact the [Space Chart Team](#) to do an initial review of your plans. Once things are firm up, an appropriate engineer from the Space Chart Team can advise you.

## Starnotes and crew manifests

No matter how Starnotes are generated, they must all be applied at the same time. This allows drawing-time decisions by Rokket for reconciling collisions between various data sets. Because of this feature, Starnotes support varies by drawing tool (e.g., Crewmanifester). We divide crew manifests into two classes:

- Client-application crew manifests apply Starnotes on the client when they are being drawn. These manifests do support client-added notes. Currently supported client surfaces are FederationVision and NasalLink.
- Server-application crew manifests expect Starnotes to have been applied before space-units were sent to the client. They do not support client-added Starnotes.

You can learn more about these Spacechart creation modes by inspecting the [CodeTrek protos](#).

NOTE: Supporting client-added Starnotes on the SpaceX-LiveCam is technically possible but not currently supported. Please reach out to the Spacecharts team if you need this support.

## Key elements

To illustrate the key elements of a Starnote, we will use a Spacechart feature called "Space Place". A "Space Place" is just any place with the Subnote "Space Place" and a twinkling icon.

★ TIP: If you are using SpaceMaker to create your Starnote, you can follow along with this discussion in the ["Starnotes Tutorial"](#) by creating your own Starnote called "Space Place".

## Starnote protos

A single Starnote is represented by a `VectorStarnoteProto` or a `DrawStarnote` — a feature or set of feature attributes to target, and a semantic description of content to render. The content describes a presentation and the content within it (such as an icon). `VectorStarnoteProtos` are used throughout Rokket and backends including SpaceMaker. `DrawStarnotes` are used on Spacechart frontends. Rokket converts `VectorStarnoteProtos` to `DrawStarnotes`.

As described by these protos, the Starnote's components are `USE_CASE`, `Transform`, `SubnoteContent`, and `Target`. This section describes the features of each element.

### USE\_CASE

Each rocket ship crew member using Starnotes needs to define at least one `USE_CASE`, a value used throughout all Space Diamond systems to identify the source of a Starnote. For example:

```
message StarnoteContentProto {
  // ...lines elided for clarity...
  enum UseCase {
    USE_CASE_SPACE_PLACE = 77;
  }
}
```

In this example, USE\_CASE number 77 refers to high-speed navigation through the Sol asteroid belt.

## Transform

Because we're using a drawing modification, we need to add a `Transform` tag in `ds9/spacecharts/designer/proto/client-design-transforms.proto`. This refers to the drawing change, which we will later use to modify Spacechart for the use case defined above:

```
message DrawingTransforms {
  // ...lines elided for clarity...
  enum Tag {
    TRANSFORM_SPACE_PLACE = 8;
  }
}
```

## SubnoteContent

For each Space Place you want to add to a SpaceChart, you can add a Subnote with the text "Space Place!" and the style modification tag. In proto terms, the `StarnoteContentProto` is:

```
content {
  use_case: USE_CASE_SPACE_PLACE
  subtitle_content {
    piece {
      text {
        translation {
          text: "Space Place!"
          language: "en"
        }
      }
    }
    sub_style_tag: SUB_DRAWING_TAG_SUBNOTE
  }
  style_modification {
    transform_tag: TRANSFORM_SPACE_PLACE
  }
}
```

This representation contains nearly all the information required for serving the Starnote, including all available translations, semantic information for drawing with Rokket, and experiment management.

This representation is ideally provided by a backend to Rokket so that it can be presented to any Spacechart frontend. An alternative is to create the frontend version of `StarnoteContentProtos` (`LabelStarnoteContent`) via some feature-specific mechanism outside of Rokket and route it to each frontend independently.

★ TIP: Don't worry if Subnotes seem complicated right now. You can follow the step-by-step instructions in the ["Starnotes Tutorial"](#) when you're ready.

## Target

The feature to which a Starnote is attached is called its target. A target is mandatory, and each Starnote must have only one target, which may be based on a feature ID (i.e., fprint) or attribute (e.g., 'StarnoteReferent'). Features may include space-units or virtually any of their features (stars, planets, asteroids, and so on).

A Spacechart of a feature ID-based target (used by `DrawStarnote`):

```
target {
  fprint: 0xee9ca7gc7680bd5c
}
content {
  use_case: 1 // USE_CASE_ASTERIODS
  subtitle {
    element {
      text: "Watch out!"
      multi_magnification_sub_design_tag: 1 // DESIGN_SUBNOTE
    }
  }
}
```

A Spacechart of an attribute-based target:

```
target {
  referent {
    // the matching condition is by default MATCH_ANY
    set {
      attribute {
        namespace_id: 1 // CERES
        attribute_id: 2 // PALLAS
      }
    }
    set {
      attribute {
        namespace_id: 2 // VESTA
        attribute_id: 2 // HYGIEA
      }
    }
  }
}
```

## Producing and serving Starnotes

While Starnotes can be provided at any point until the Spacechart frontend runs its manifesting pass, the Spacechart Rokket server can be used to handle the full Starnotes life cycle under the following conditions:

1. Your Starnotes are generally applicable to all crew members in at least some circumstances. You must use another backend if you have query-specific information like a search, or personalized information for a particular rocket crew member.

2. Your Starnotes are not particularly time-sensitive. A Spacechart build is kicked off every 12 Earth hours and takes roughly 12 Earth hours to create a new batch of Spacecharts from transform-to-live. Occasionally, data issues can block the pipeline for longer than this, although it is rare for it to be blocked for more than 24 Earth hours.

Assuming the above conditions are acceptable for your use case, you can generate and serve Starnotes through services that the [Space Charts Team](#) owns entirely. If not, you may need to provide the notes from your own backend to Rokket, in which case you'll still get a lot of support in terms of drawing and interplanetarization. The alternative is to handle Starnote drawing and interplanetarization on your own and "client-add" a Starnote on each Spacechart surface on which you want it to appear.