

Manage marked text with custom IDs

The Angular extractor generates a file with a translation unit entry each of the following instances.

- Each `i18n` attribute in a component template
- Each `$localize` tagged message string in component code

As described in [How meanings control text extraction and merges](#), Angular assigns each translation unit a unique ID.

The following example displays translation units with unique IDs.

When you change the translatable text, the extractor generates a new ID for that translation unit. In most cases, changes in the source text also require a change to the translation. Therefore, using a new ID keeps the text change in sync with translations.

However, some translation systems require a specific form or syntax for the ID. To address the requirement, use a custom ID to mark text. Most developers don't need to use a custom ID. If you want to use a unique syntax to convey additional metadata, use a custom ID. Additional metadata may include the library, component, or area of the application in which the text appears.

To specify a custom ID in the `i18n` attribute or `$localize` tagged message string, use the `@@` prefix. The following example defines the `introductionHeader` custom ID in a heading element.

The following example defines the `introductionHeader` custom ID for a variable.

```
variableText1 = $localize :@@introductionHeader:Hello i18n! ;
```

When you specify a custom ID, the extractor generates a translation unit with the custom ID.

If you change the text, the extractor does not change the ID. As a result, you don't have to take the extra step to update the translation. The drawback of using custom IDs is that if you change the text, your translation may be out-of-sync with the newly changed source text.

Use a custom ID with a description

Use a custom ID in combination with a description and a meaning to further help the translator.

The following example includes a description, followed by the custom ID.

The following example defines the `introductionHeader` custom ID and description for a variable.

```
variableText2 = $localize :An introduction header for this sample@@introductionHeader:Hello i18n! ;
```

The following example adds a meaning.

The following example defines the `introductionHeader` custom ID for a variable.

```
variableText3 = $localize :site header|An introduction header for this sample@@introductionHeader:Hello i18n! ;
```

Define unique custom IDs

Be sure to define custom IDs that are unique. If you use the same ID for two different text elements, the extraction tool extracts only the first one, and Angular uses the translation in place of both original text elements.

For example, in the following code snippet the same `myId` custom ID is defined for two different text elements.

The following displays the translation in French.

Both elements now use the same translation (`Bonjour`), because both were defined with the same custom ID.

@reviewed 2021-10-13