salesforce VRP Consulting

Lecture: Custom Settings and Custom Metadata Types.

Goryachko A.
November 19, 2018

Custom Settings

Custom settings are similar to custom objects. Application developers can create custom sets of data and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. Formula fields, validation rules, flows, Apex, and the SOAP API can then use this data.

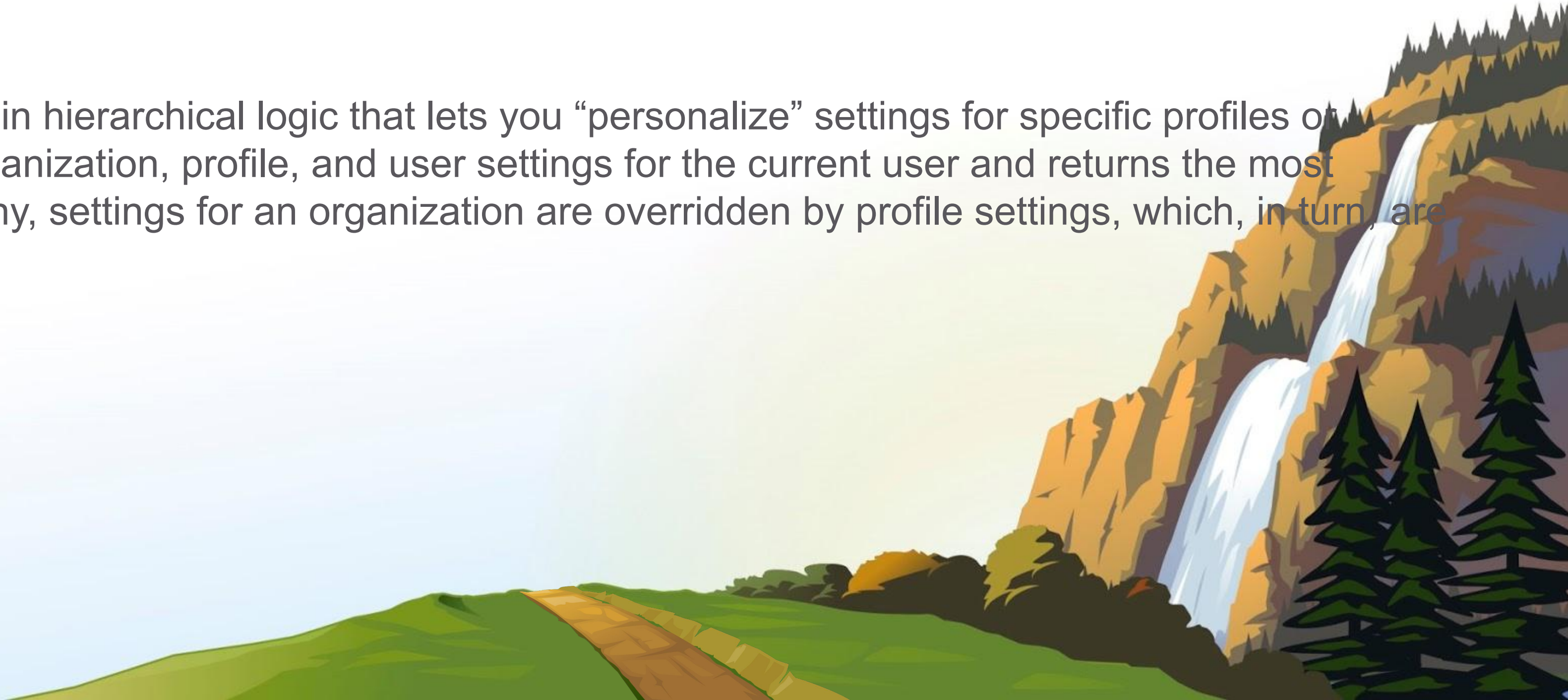
Types of Custom Settings

- ## List Custom Settings

  A type of custom setting that provides a reusable set of static data that can be accessed across your organization. If you use a particular set of data frequently within your application, putting that data in a list custom setting streamlines access to it. Data in list settings does not vary with profile or user, but is available organization-wide. Examples of list data include two-letter state abbreviations, international dialing prefixes, and catalog numbers for products. Because the data is cached, access is low-cost and efficient: you don't have to use SOQL queries that count against your governor limits.

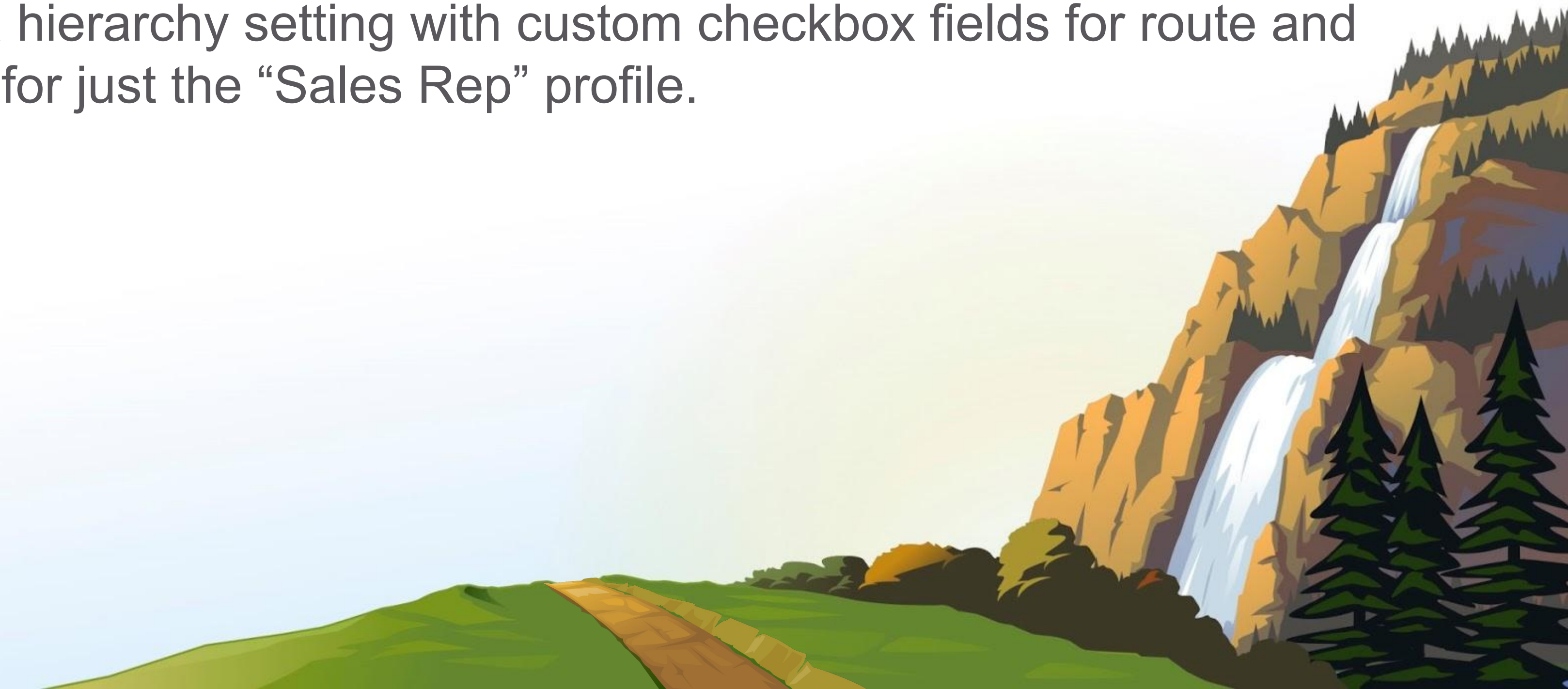
- ## Hierarchy Custom Settings

  A type of custom setting that uses a built-in hierarchical logic that lets you “personalize” settings for specific profiles or users. The hierarchy logic checks the organization, profile, and user settings for the current user and returns the most specific, or “lowest,” value. In the hierarchy, settings for an organization are overridden by profile settings, which, in turn, are overridden by user settings.



When we should use Custom Settings?

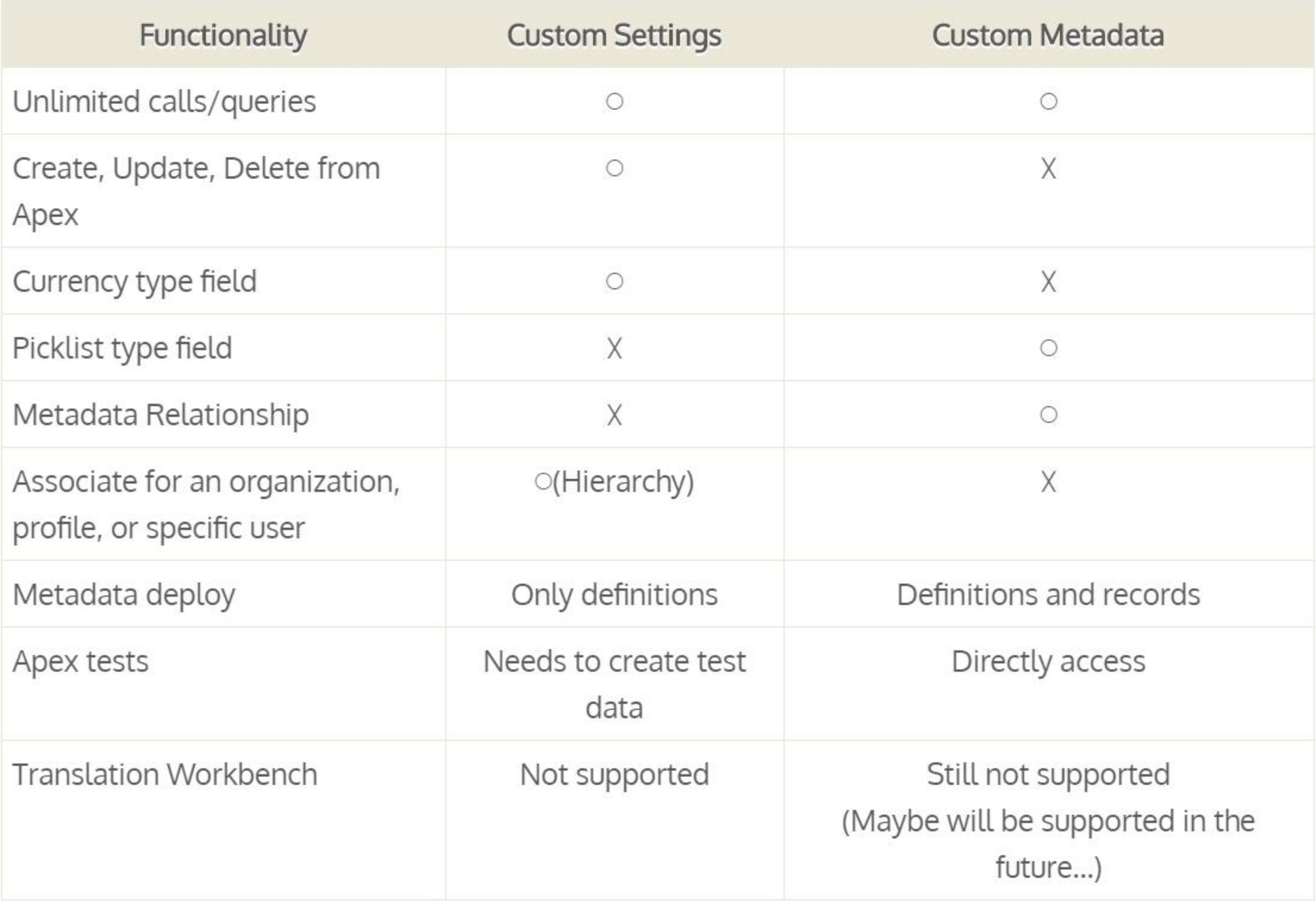
- A shipping application requires users to fill in the country codes for international deliveries. By creating a list setting of all country codes, users have quick access to this data without needing to query the database
- An application displays a map of account locations, the best route to take, and traffic conditions. This information is useful for sales reps, but account executives only want to see account locations. By creating a hierarchy setting with custom checkbox fields for route and traffic, you can enable this data for just the “Sales Rep” profile.



Custom Metadata Types

Custom metadata is customizable, deployable, packageable, and upgradeable application metadata. First, you create a custom metadata type, which defines the form of the application metadata. Then you build reusable functionality that determines the behavior based on metadata of that type.

Differences between Custom Metadata and Custom Settings



Functionality	Custom Settings	Custom Metadata
Unlimited calls/queries	○	○
Create, Update, Delete from Apex	○	X
Currency type field	○	X
Picklist type field	X	○
Metadata Relationship	X	○
Associate for an organization, profile, or specific user	○(Hierarchy)	X
Metadata deploy	Only definitions	Definitions and records
Apex tests	Needs to create test data	Directly access
Translation Workbench	Not supported	Still not supported (Maybe will be supported in the future...)

Advantages to Custom Metadata

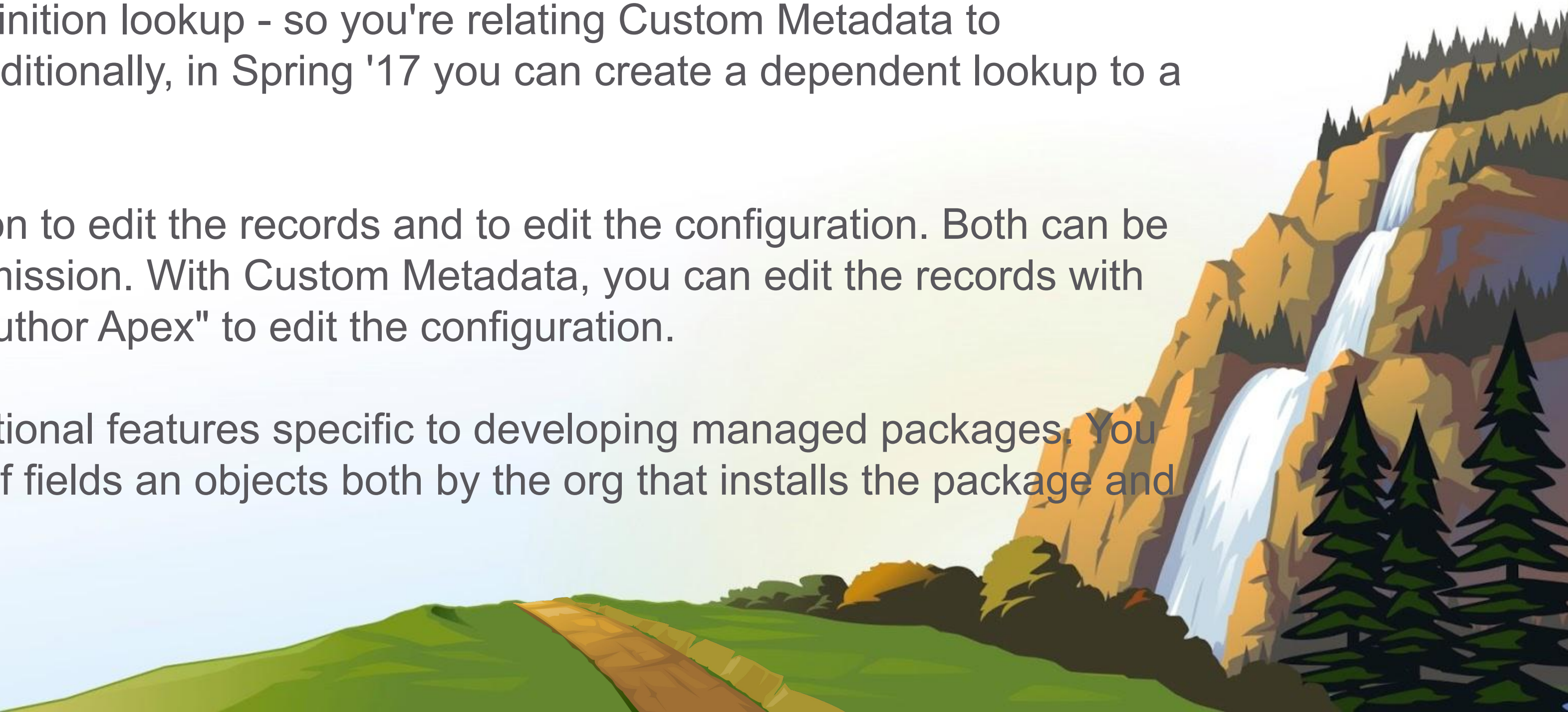
- Most importantly, as stated before, they are Metadata and thus deployable! No more annoying configuration after deployment, which you have to do with Custom Settings. They're also refreshed to sandboxes, so you don't need to create Apex classes to generate your default Custom Setting records.

- They have WAY more options that Custom Settings: picklist fields, long text areas (in Spring '17), page layouts, and validation rules

- The beauty that is Metadata Relationships! You can create lookups between Custom Metadata objects. Additionally, you can create an Object Definition lookup - so you're relating Custom Metadata to Standard or Custom Object definitions. Additionally, in Spring '17 you can create a dependent lookup to a Field Definition on that object.

- Custom Settings have the same permission to edit the records and to edit the configuration. Both can be done with the "Configure Application" permission. With Custom Metadata, you can edit the records with "Configure Application" but you require "Author Apex" to edit the configuration.

- Custom Metadata types have a lot of additional features specific to developing managed packages. You can configure the visibility and editability of fields an objects both by the org that installs the package and by upgrades to the package.

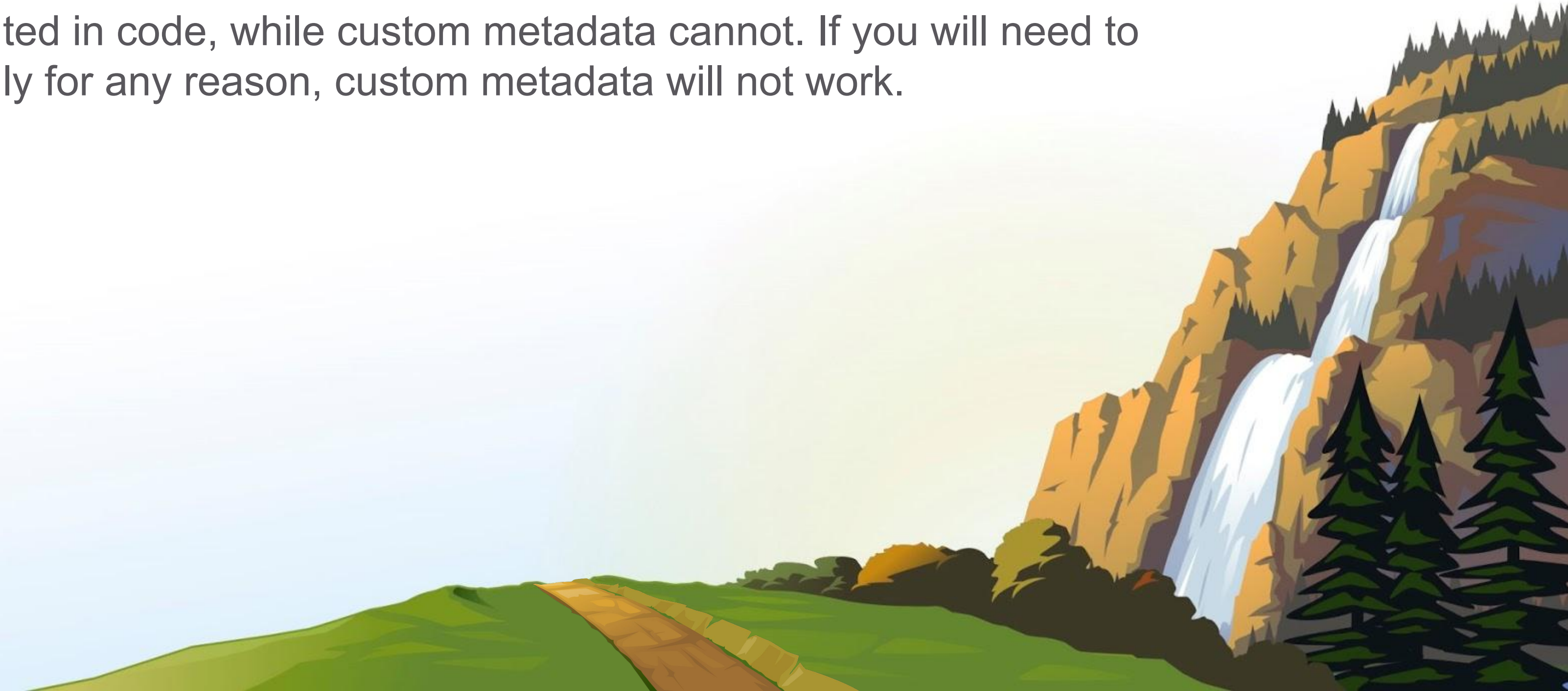


Reason why you would use custom settings instead

- Hierarchies - Custom metadata types do not purport to replace hierarchy custom settings which allow you to vary values based on users and profiles across the org. These custom settings can also be referenced in formulas, so can be used in formula fields, validation rules, workflow rules, and visualforce.

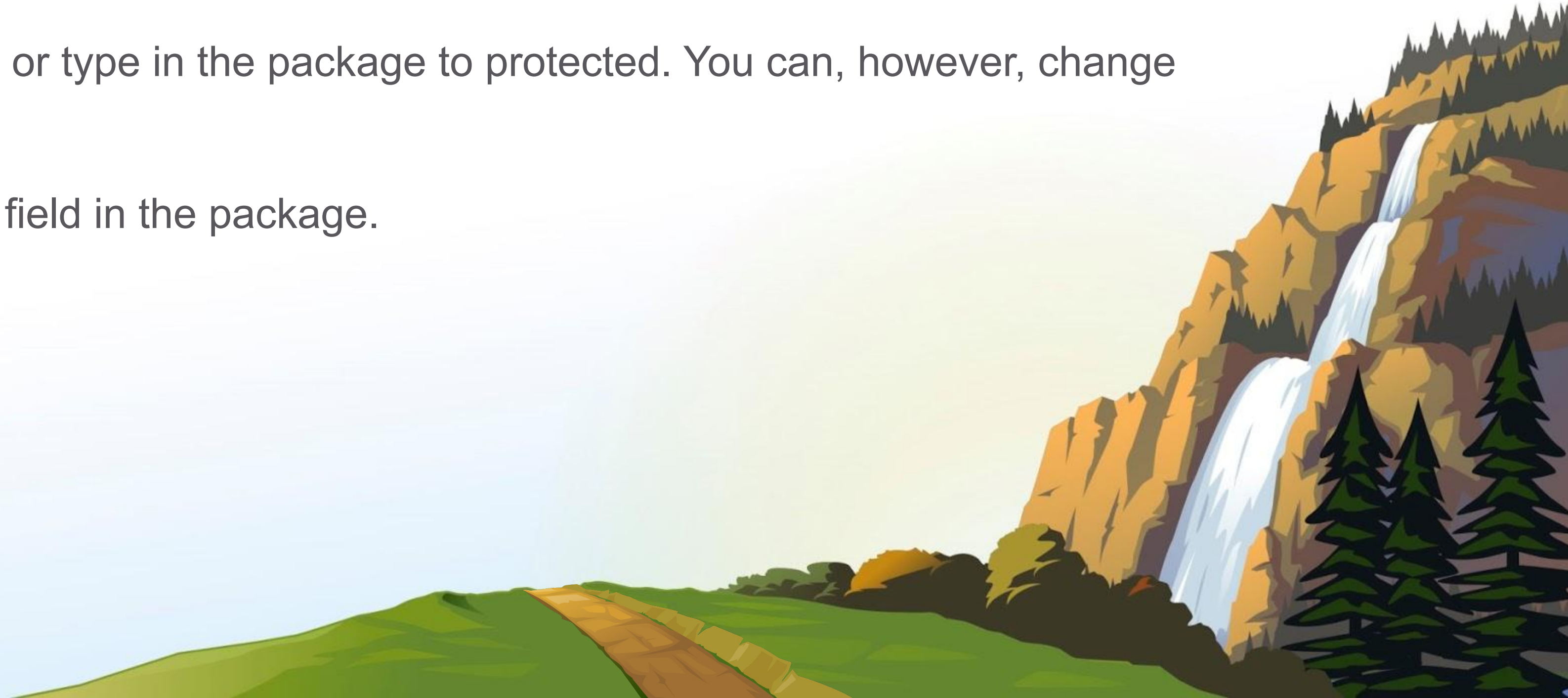
- Web service credentials - If you’re using test credentials in a sandbox, you don’t have any reason to deploy them to production. More importantly, if you create a sandbox, you don’t want the config for the production versions of your Web Services being created automatically and used by default. However, it is usually best to use Named Credentials for this.

- Custom setting records values can be edited in code, while custom metadata cannot. If you will need to modify your configuration programmatically for any reason, custom metadata will not work.



Considerations for Custom Metadata Type Packages

- Add required fields to the custom metadata type.
- Set non-required fields to required.
- Delete custom fields that are in the uploaded version of the package. If you add a custom field after the upload, you can still delete it until you upload a new Managed - Released version.
- Delete public custom metadata records that are included in the package.
- Change a public custom metadata record or type in the package to protected. You can, however, change protected records and types to public.
- Change the manageability of any custom field in the package.



Links

Custom Settings Methods:
https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_methods_system_custom_settings.htm
Custom Settings:
https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_customsettings.htm
Trailhead Custom Metadata Types:
https://trailhead.salesforce.com/en/content/learn/modules/custom_metadata_types
Custom Metadata Types:
https://help.salesforce.com/articleView?id=custommetadatatypes_overview.htm&type=5