**Packages**

A Salesforce **package** **is a container that holds a specific set of metadata elements**. It can be installed into an org to add or modify functionality.

Main idea: packages offer a convenient way   
 - to **bundle and distribute application functionality**,   
 - making it **easier to deploy**   
 - and **manage customizations across orgs**.

всего 3 типа пакетов версии  
 - managed v1 v2  
 - unmanaged v1   
 - unlocked v2

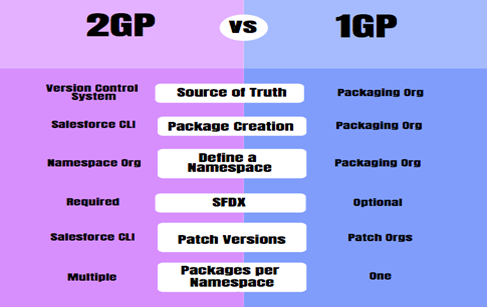
V1 – before Salesforce DX  
 unmanaged – one-time deployment  
 manged – for SF partner that build and list apps on AppExchange ISVs (Independent Software Vendors)

V2 – with SFDX

Номера версий форматируются как  **minor.patch.build**

*Metadata Coverage*

https://developer.salesforce.com/docs/metadata-coverage/62



**Difference between unmanaged packages, managed packaged, and unlocked packages.**

|  |  |  |
| --- | --- | --- |
| **Unmanaged Package** | **Managed Package** | **Unlocked Package** |
| Not upgradable   нужно удалить пакет и установить его заново | Upgradable, and one can namespace them  1 орг = 1 уникальный неймспейс = 1 пакет V1 | Upgradable and one can namespace them |
| Metadata elements are not IP Protected  внутри пакета все можно менять (класс/компонент итд) | Metadata elements are IP Protected  нельзя менять ничего внутри пакета | Metadata elements are not locked and can be changed by system admins |
| Allows for the creation of extension packages | Can be created in salesforce UI and distributed via appexchange | Requires Salesforce CLI to generate them |
| Unmanaged package containers automatically pull dependency | Components are locked and one cannot modify them directly in production or sandbox | Allows you to build your applications in a modular way |
|  | It is easier to manage when codebase is modularized | It is easier to manage when the codebase is modularized |

**Managed Packages**

https://www.youtube.com/watch?v=N1IcZcSvWmE&ab\_channel=SalesforceDevelopers

**Managed Packages** - They are typically used for distributing and selling applications to customers, providing a way to protect intellectual property and control upgrades.

NameSpace  
  
@namespaceAccessible анотация

**Unlocked Packages**

https://www.youtube.com/watch?v=oY-dHAVJm34&t=2052s&ab\_channel=SalesforceApexHours

Unlocked Packages are part of Salesforce’s package-based development model. They allow developers to package metadata and business logic to simplify deployment processes and enable better release management.

Unlocked packages **follow a source-driven development model**. The **source of truth** of the metadata contained within the package **is your version control system**, not what’s in an org.

**Modularity:** Unlocked Packages allow for granular control over the components included, enabling developers to build and distribute modular features. This modularity promotes code reusability and makes it easier to manage and maintain codebases.

**Dependency management:** Developers can define dependencies between packages, ensuring that all required components are included during installation. This feature simplifies the deployment process and reduces the risk of missing dependencies.

**Accessibility**: Unlike managed packages, unlocked packages provide access to the source code, allowing for easier customization and extension of packaged functionality. This level of accessibility empowers developers to tailor the package to their specific needs and integrate it seamlessly with existing codebases.

***Allows:***  
 - to organize their existing metadata   
 - package an app

- extend an app that they’ve purchased from AppExchange   
 - package new metadata

-helps you to add, edit, and remove metadata in your org in a trackable way

#### Benefits

**Simplified development**: Unlocked packages promote modular development practices, making it easier to manage and maintain code. Developers can focus on building individual features without worrying about the intricacies of the entire application.

**Incremental release**: With the ability to define dependencies, developers can release updates to individual packages without impacting the entire application. This incremental release approach allows for faster and more efficient development cycles.

**Collaboration**: Unlocked packages facilitate collaboration by allowing multiple developers to work on different packages simultaneously. Each developer can focus on their specific area of expertise, leading to faster development and increased productivity.

***What are the benefits of packaging over Change Sets and ANT Migration Tool***

Packages promote a source-driven development approach and version control system. With package versions, you have an **immutable**, **versionable** **artifact** that can be used in CI.

With a version-based approach, you can develop and install different package versions in target organizations and **adopt a build-once, deploy-anywhere approach**.

You can express dependencies among packages. With this capability, you can embrace a modular approach where each package contains a set of metadata that **represents a distinct unit of functionality**, with well-defined dependencies among packages.

Packages provide a **repeatable, scriptable, and trackable** way of managing change as you develop functionality **using Salesforce DX.**

***Steps*** 1 we need DevHub org  
 2 add metadata to package  
 3 create package (get package id 0Ho…..)  
 *sf package create --name object-module --package-type Unlocked --path object-app -d "Objects and Permission for them"*

***3a add dependencies if need "dependencies":***

4 create package version (beta) allow to install package (get version Id 04t… )  
 *sf package version create -p object-module -d object-app -w 10 -x*

5 Promote package – allow install on Prod

*sf package version promote -p 04tQy0000003mPtIAI****@1.0.0-1*** *-n*

sf package install -p 04tQy0000003lgjIAA  
  
script for instalation dependencies https://developer.salesforce.com/docs/atlas.en-us.sfdx\_dev.meta/sfdx\_dev/sfdx\_dev\_unlocked\_pkg\_install\_pkg\_sample\_script.htm

ScratchOrg Snapshot – позволяет настроить орг и запомнить настройки Все последующие орги создаются по этому шаблону (с предустановленными пакетами)

### ***Tips for Maximizing Unlocked Packages***

***Plan your package structure:***Design a modular package structure to ensure efficient management and distribution of features. By breaking down your package into smaller, more manageable components, you can easily update and maintain specific functionalities without affecting the entire package.

***Version control system:*** Utilize version control systems like Git to track changes and manage different package versions. This will enable you to keep a record of all modifications made to your package, making it easier to roll back changes if necessary.

***Document everything:***Document package contents, dependencies, and installation instructions to facilitate collaboration and onboarding of new team members. Clear and comprehensive documentation will ensure that everyone involved in the project understands how to work with the package and its associated components.

**Unmanaged Packages**

Unmanaged packages are the traditional packaging option in Salesforce. They provide a straightforward approach to bundle and distribute customizations, but lack certain features and flexibility offered by unlocked packages.

**Few unique characteristics**

**Fixed dependencies:** Once components are included in an unmanaged package, their dependencies are fixed, making it challenging to manage complex relationships.

**No source code access:** Unmanaged packages do not provide access to the source code, limiting customization options for packaged features.

**Complete control:** Organizations using unmanaged packages have full control over the components included and can modify them directly in the target org.

***Beneficial in certain scenarios:***

***Quick deployment:****Unmanaged packages are easy to create and deploy, making them suitable for quick customization transfers.*

***Ongoing customization:****Organizations that need to modify packaged components directly in the target org may find unmanaged packages more suitable.*

### ***Guidelines for Utilizing Unmanaged Packages***

***Keep customizations minimal:***Minimize modifications to packaged components to reduce the risk of conflicts and compatibility issues during package upgrades. While it may be tempting to customize the package to fit your specific needs, it’s important to strike a balance between customization and maintainability.

***Back up target orgs:*** Before deploying unmanaged packages, create backups of target orgs to ensure data integrity and easy rollback in case of issues. This precautionary step will safeguard your data and allow you to quickly revert to a previous state if any problems arise during the deployment process.

***Detailed testing:*** Perform comprehensive testing to ensure that modifications made through unmanaged packages do not negatively impact the existing functionality. Thoroughly test all changes and customizations to ensure that they integrate seamlessly with your existing Salesforce setup.