
Partner’s Guide to Odoo Upgrades

Release 0.1.0

Olivier Laurent (OLT), Odoo SA

Oct 25, 2019

Contents

1	Introduction	2
2	FAQ	3
2.1	What is an upgrade ? What is not an upgrade?	3
2.2	What’s the whole process from the first upload to the final upgrade?	3
2.3	Can we upgrade a database to the community version?	4
2.4	Can we only upgrade parts of a database or can we deliver a partialy upgraded database?	4
2.5	Can you give the customer an ETA?	4
2.6	When are the upgrade scripts for a new Odoo version released?	6
2.7	Is the upgrade queue really based on a “first come, first served” basis or can some customers/databases be prioritized?	6
2.8	Is it possible to upgrade directly from v10.0 to v12.0?	6
2.9	Can you still upgrade very old Odoo versions like 6.0?	6
2.10	Do you have stats about the Upgrade requests?	6
2.11	Do you advise to upgrade the customer each year or is it better to upgrade him every 2 years?	6
2.12	Can we install (or uninstall) new modules during the course of an upgrade project?	6
2.13	Can we modify our custom modules during the course of an upgrade project?	7
2.14	Technically speaking, how do you upgrade a database?	7
3	What to do when the database is finally delivered?	8
3.1	Restore the upgraded database	8
3.2	Copy your old filestore	8
3.3	Make your custom modules available	8
3.4	Perform an update of your modules	9
4	Best practices	10
4.1	Always follow the same process	10
4.2	Don’t change the database structure or the Apps configuration	10
4.3	Don’t change the database Unique ID	11
4.4	Do not plan the final upgrade until all issues have been fixed	11
4.5	Test your database thoroughly	11
4.6	Make a last test a week before the final upgrade	11
	Index	13

1 Introduction

The intended audience of this document is the Odoo partners and Odoo customers with sufficient *system administration* skills to handle an On Premises' database upgrade.

This guide mostly focus on On Premises databases. Some sections are not suited to Odoo.sh databases.

2 FAQ

2.1 What is an upgrade ? What is not an upgrade?

An upgrade is the process of converting an Odoo database from one version to a higher version.

The original database version is called the **source** version.

The upgraded database version is called the **target** version.

So, if the source database is **8.0** and the target version is **10.0**, this is an upgrade.

Here are some examples of **non upgrades**:

Source version is 10.0, target version is 10.0

- this is not an upgrade: the target version should be strictly higher than the source version
- this is just a relocation from one server to another or a simple update of 'base' module

Source version is Community 10.0, target version is Enterprise 10.0

- nearly the same case as the previous one
- it's just a matter of installing the Enterprise modules
- see <https://www.odoo.com/documentation/12.0/setup/enterprise.html>

Source version is 10.0, target version is 8.0

- this is not an upgrade: the target version should be strictly higher than the source version
- this is actually called a *downgrade*
- this is quite difficult to achieve

Source database is not Odoo, target is Odoo 10.0

- example: Source database is from another software (Compiere, Openbravo, MS Dynamics, ...)
- this is not an upgrade: we only upgrade Odoo databases not other software's databases

Source version is 4.2, target version is 10.0

- technically speaking, this is an upgrade but the minimum source Odoo version should normally be **5.0**.
The reason: the Odoo Upgrade Services started in 2011 when we released Odoo version **6.0**, so we have no upgrade scripts for target version **5.0** (or lower)

2.2 What's the whole process from the first upload to the final upgrade?

An upgrade project is a two-step process:

1. one or more test upgrades
2. the final upgrade

The first step is the longer one. The second step can take less than an hour to perform.

When a database is firstly uploaded, the upgrade platform (upgrade.odoo.com) will try to upgrade it automatically.

If the database cannot be upgraded automatically, the Upgrade team will fix it by writing specific upgrade scripts.

This step can take from several days to a few weeks or months days depending on the complexity and the number of problems.

If the automatic process succeeds, the Upgrade team will perform a series of functional tests on it.

Issues found during this testing phase will be fixed and, when there are no issues anymore, we'll deliver an upgraded test database. The customer will then have to thoroughly test the database.

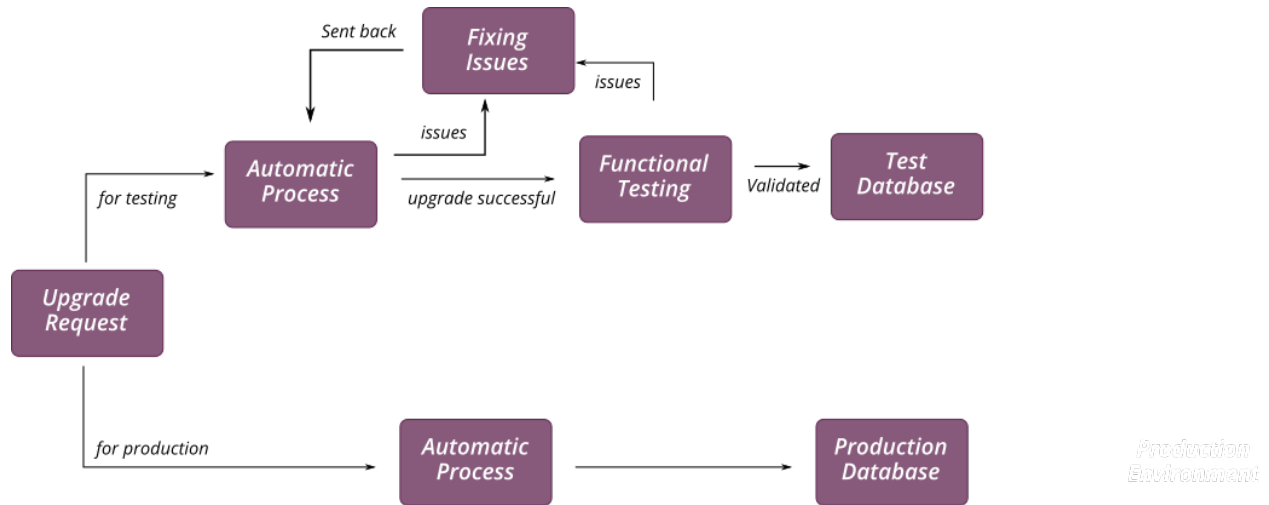
This is an important step because we only test the standard flows. If specific flows are used, the Upgrade team cannot test them because they don't know the way the users are using the database.

If the customer discovers some issues, he'll be able to submit them to us (www.odoo.com/help), and they will be handled as usual. The Upgrade team will fix them using the upgrade scripts, then relaunch the upgrade process and the already uploaded database will be upgraded using the new upgrade scripts.

During all this time, the customer still need to continue working as usual on his old version of Odoo.

Once the customer is satisfied with the result, he can plan the final upgrade. This time the customer stops working on his production database, he creates a backup and send it on the Upgrade platform and it will be migrated right away without any testing phase. Most of the time, databases are migrated in less than an hour.

The process can be summarized as:



2.3 Can we upgrade a database to the community version?

No.

We always upgrade databases (Online or On Premises) using the enterprise addons. And so, the end result will always be an Enterprise database.

The customer is, of course, free to uninstall the Enterprise addons, but Odoo won't offer any help (no support) regarding this.

2.4 Can we only upgrade parts of a database or can we deliver a partially upgraded database?

No.

We always upgrade a database as a whole. We can't choose which modules will be upgraded and which one will not be upgraded.

So, we can't skip the Accounting section for example.

And we don't deliver partially upgraded databases. Either a database is totally upgraded or it's not.

2.5 Can you give the customer an ETA?

No.

It's impossible to know how much time we'll take to upgrade a database.

The process to upgrade a database is linear. We don't know in advance how many issues we'll have to fix. It could be 0, it could be 10. It could be more.

The process is as follows:

1. we launch the upgrade process
2. if it fails, we fix the issue and re-launch the upgrade process -> go to 1.
3. if it succeeds, it's OK, we have the upgraded database -> end

So, if there are 5 issues to fix, the process will be:

```
1 -> 2 -> 1 -> 2 -> 1 -> 2 -> 1 -> 2 -> 1 -> 2 -> 1 -> 2 -> 1 -> 3
```

But the very first time we launch the process, we don't know if we'll get an issue or not.

And when we fix the first issue, we don't know if we'll get a second one. It's only at the end of the process, when the database is finally upgraded that we know how many issues we had to fix. Only one quarter of the databases are upgraded right away, without any custom fixes.

The number of issues is not the only factor that can impact the time to get an upgraded database. The complexities of these issues is also an important factor.

Some issues are simple to fix. Some issues are quite complex, and require several teams to be fixed.

Another important factor is the size of certain specific tables in the database.

The size of a database is not necessarily an important factor but if a database has millions or 10th of a million records in some tables (accounting entries or stock moves for example), the database can take ages to be upgraded and the standard scripts will have to be optimized.

Before this optimization, some databases cannot be upgraded. We need to stop the process because it's still running after several days, and we don't know if it will still continue for several days more or not. A process should, at most, take 24 hours. FYI, the average process time to upgrade a v10 database to v11 is 20 minutes.

The number of customizations and their complexities is, of course, very important too. Especially for SaaS databases (we don't work on the customizations of On Premises databases).

The last important factor is the work load of the Upgrade Team.

Most databases can be upgraded in a day or two, but we need to distinguish the actual work from the waiting period before a developer actually starts the work. The later can be quite long at time, especially right after a new Odoo release because a lot of customers try to upgrade their database.

And of course, the standard upgrade scripts for the given target version need to be ready. Without them, we cannot even start the project (see question *When are the upgrade scripts for a new Odoo version released?*)

Until then, we can accumulate a lot of databases in the pipe which will put a huge burden on the Upgrade team's work load when the scripts will be finally released.

Fortunately, this process is now well-rounded, and we don't need to wait several months before the upgrade scripts are finally released.

So, as you can see, it's impossible to guess the time it will take to have a database upgraded. We can only give the current average time.

For example, in September 2018, we delivered a database in 12 days (on average). And in August 2018, it was 11 days. We have some stats here: <https://upgrade.odoo.com/database/stats>

Here are the questions you should ask yourself to know if a database will be quickly delivered or not:

- are the upgrade scripts for the target version already released?
- is the target version released only a few months ago?
- are the Upgrade team under heavy load?
- is the size of some tables huge (millions of records)?

- is it a heavily customized Online (SaaS) database?

If you answer *yes* to one of these questions, chances are that it will not be delivered very quickly. Instead of a few weeks (2 or 3), it will mostly be a few months.

2.6 When are the upgrade scripts for a new Odoo version released?

For v12.0, we released the scripts in less than 2 months (54 days)

Way better than for v11.0 (6 months).

And better than for the previous Odoo releases (<v11.0) which was 3 months on average.

We'll try to achieve an even better release cycle for v13.0.

2.7 Is the upgrade queue really based on a “first come, first served” basis or can some customers/databases be prioritized?

It's “*first come, first served*”

But there are a few exceptions depending on the situation.

For example, if the customer really needs the new version for a legal reason.

Or if a functional consultant is blocked in his Quickstart project.

If a given database is submitted several times, the first request will still serve as a basis for the position in the queue.

A new upgrade request will not invalidate the position in the queue.

But we'll work on the last request because the data is more up to date.

2.8 Is it possible to upgrade directly from v10.0 to v12.0?

Yes.

There is no need to do the intermediary upgrades. The Upgrade platform will do it for you.

2.9 Can you still upgrade very old Odoo versions like 6.0?

Yes.

The lowest Odoo source version we still accept is 5.0

It's quite exceptional, but we still receive some 5.0 or 6.0 databases from time to time.

2.10 Do you have stats about the Upgrade requests?

Yes we have some stats directly on the [Upgrade platform](#)

2.11 Do you advise to upgrade the customer each year or is it better to upgrade him every 2 years?

Every 2 years is better but it mostly depends on the way customer handles changes.

The main question is “*Do they really need a feature available in the new version?*”

An upgrade involves a lot of commitment and work from a customer or partner's perspective.

It's also a lot of work from Odoo's perspective. A lot of people from different teams can be involved.

2.12 Can we install (or uninstall) new modules during the course of an upgrade project?

No.

You are, of course, free to do it but it can have bad consequences for you upgrade project.

- the upgrade process can fail
the upgrade scripts are heavily dependent on the database structure which itself is dependent on the state of the installed modules
if you change the database structure, the process can fail, and we'll have to analyze your database again and write additional specific scripts
- it invalidates our testing session
we test a database based on the installed modules. And we only test it once when the first request is successful.
if you install new modules, we will not test the database based on them.
- it also invalidates your own testing session
if you already have tested your database and you install new modules, you'll need to test if these modules have been correctly upgraded

2.13 Can we modify our custom modules during the course of an upgrade project?

No. For the same reason as stated on the previous answer (*Can we install (or uninstall) new modules during the course of an upgrade project?*)

If you have custom modules and you modify them during the course of the upgrade project, the result will be the same as installing new modules. It modifies the database structure.

2.14 Technically speaking, how do you upgrade a database?

A database is upgraded “*in place*”.

No export or import. No new database is created with stuff copied in it. We don't copy data from one database to another.

We take the same database in its original version and perform a “module update” using the next higher Odoo version. We apply some scripts before that update and some scripts after (what we call “pre” scripts and “post” scripts).

Ex:

- we take a v8 database
- execute the standard pre upgrade scripts written for v9
- perform an “all modules update” (“base” module -> Upgrade button) using Odoo v9
- execute the standard post upgrade scripts written for v9
- -> we get a v9 database

Since the database is upgraded “*in place*”, objects that have not changed since the last Odoo version are not touched. They are still there. We don't modify them.

3 What to do when the database is finally delivered?

3.1 Restore the upgraded database

Use the Odoo Database Manager to restore the upgraded zip file. The *Odoo Database Manager* is usually available on the `/web/database/manager` URL of your Odoo server

3.2 Copy your old filestore

During the upgrade process, we sometimes convert some binary fields to *ir.attachment* records. We also sometimes directly create *ir.attachment* records like assets for example.

That's why we send you a zip file containing the upgraded dump and a filestore folder containing these new files.

Your old Odoo version was probably using a filestore. That filestore stayed at your place. You have not send it to us (because we have no use of it and it could be quite heavy, slowing down the upload of you dump)

So, when you receive the upgraded database, you'll need to merge the old filestore (that stayed on your server) with the new filestore (contained in the upgraded zip file)

Odoo normally stores the filestore at:

```
/home/ODOO_USER/.local/share/Odoo/filestore/DB_NAME
```

But it's not always the case. Your filestore could be elsewhere depending on your setup.

One way of finding where your filestore is located is by using these commands:

If your old Odoo version is 10.0 or higher:

```
cd /path/to/old/odoo
python -c "from odoo.tools import config;print(config.filestore('your_old_db_name'))"
```

If your old Odoo version is 9.0 or lower:

```
cd /path/to/old/odoo
python -c "from openerp.tools import config;print(config.filestore('your_old_db_name
↳'))"
```

Just replace "*your_old_db_name*" by the database name of you actual installation.

Be sure to use the right python version. Odoo v11.0 and higher uses *python3*

When you have determined where your old filestore is located, you'll need to copy it to the new location.

A command similar to this one:

```
rsync -avi /path/to/filestore/old_db/ /path/to/filestore/new_db/
```

will do.

3.3 Make your custom modules available

The database you receive will have your custom modules in the "*to upgrade*" state. So, you'll have to perform a *module update* to have them upgraded to the new version.

The 2 requirements are:

- they should be available in their new version
- they need to be available in the `--addons-path` option of your Odoo server

3.4 Perform an update of your modules

This operation is required if you have custom modules installed because they will be in the *to upgrade* state in the upgraded dump file. Your custom modules should, of course, be available in the new Odoo version.

The module update is usually performed using these command line options:

```
-d DBNAME  
-u all  
--stop-after-init
```

4 Best practices

4.1 Always follow the same process

The process is the set of actions you do before uploading the database on the upgrade platform and the actions you'll have to perform after downloading the upgraded database.

Testing the process is as important as testing the database.

Depending on your case, the process can be very simple if you have a simple setup and a small database with not a lot of customization or quite complex if your database is huge and you have to perform operations before and after the upgrade.

As soon as you have a process that works, don't change it or think twice if you really need to change it. If you change it, test it again.

The most basic actions are:

- create a database dump
- upload it on the upgrade platform

Then:

- download the upgraded database
- restore it on the new Odoo environment

Partners sometimes need to perform additional actions either before or after the upgrade. Like:

- create a copy of the production database and execute some SQL queries before uploading it
- execute upgrade scripts for their own custom modules
- ...

If your process is not scripted and so, all manual, document it thoroughly but it's probably better to write a script that will execute these actions automatically.

You can use the Upgrade API to help you in this process. It's documented here:

```
https://www.odoo.com/documentation/12.0/webservices/upgrade.html
```

You can find a sample script using this API here:

```
https://gist.github.com/olilau/6343c51b8f4452d7ce66a962f09cf674
```

Here is an example of use:

```
CONTRACT=M1234-abcd EMAIL=abc@example.com AIM=test TARGET=10.0 DUMP=/path/to/dbdump.  
↪sql.gz bash odoo_upgrade_request.sh
```

4.2 Don't change the database structure or the Apps configuration

Avoid installing (or uninstalling) modules, adding or modifying views or changing the settings of the database during the course of an upgrade project.

The upgrade scripts are heavily dependent on the database structure which itself is dependent on the state of the installed modules.

If you change the database structure, the upgrade process can fail, and we'll have to analyze your database again and write additional specific scripts, which will postpone the delivery of the database.

It also invalidates our testing session.

We test a database based on the installed modules. And we only test it once when the first request is successful. If you install new modules, we will not test the database based on them.

And it also invalidates your own testing session.

If you already have tested your database and you install new modules, you'll need to test if these modules have been correctly upgraded.

4.3 Don't change the database Unique ID

The Unique ID of your database can be found with the following SQL query:

```
select value from ir_config_parameter where key = 'database.uuid';
```

It's very important that you don't change it before uploading your database. We use this unique ID to group all your upgrade requests and to know if a new upload has already been tested or not. We never deliver a database if it has not been tested before by our team, except if explicitly requested.

If you change this unique ID, your upload will always end up in Testing stage and will not be delivered automatically. This is very important for your final upgrade. If you change this ID, the database that you upload for your final upgrade will end up in Testing stage and will only be delivered manually by our team. For example, if you upload it Friday evening, it will only be delivered to you Monday morning.

4.4 Do not plan the final upgrade until all issues have been fixed

If you have pending issues that we need to fix, do not schedule the final upgrade. Wait until they are all fixed.

Issues encountered post-upgrade are trickier to solve and can block your business processes.

4.5 Test your database thoroughly

A rushed upgrade is seldom a good idea, often a source of frustration.

Testing is the most important phase of your migration and requires that you commit time to it. No one more than you can know if something is wrong in your processes and business flows.

We advise testing at the very least the following things (depending on what apps you use):

Usual flows test all the flows that you or your colleagues/employees usually use.

Reporting data compare reporting data in your test database to your current production database.

Customizations if you have customized your Odoo instance with Odoo Studio or through an Odoo Success Pack, test all the changes that have been made to your Odoo database.

Website make sure all the public pages of your website look as they should.

General testing open form views, explore your database as much as possible to ensure that all screens work properly.

This is a non-exhaustive list that you can expand based on your usual work in Odoo.

Although the Odoo Upgrade service does test your database, you are the only person able to tell if everything is as it should be. Make sure that your testing is as complete as possible to avoid issues after the migration of your production database.

4.6 Make a last test a week before the final upgrade

Although we try to ensure the process will be as smooth as possible for your final upgrade, we can never be 100% certain that a change in Odoo source code or the upgrade scripts will not break the process.

It happens very rarely but it could still happen.

One way to minimize this risk is to upload a last test database a week or so before the actual final upgrade. In case something bad happens, we'll have time to fix it.

Index

B

best practices, 10

C

community version, 4

custom modules, 7, 8

D

database structure, 10

database.uuid, 11

downgrade, 3

E

ETA, 4

F

filestore, 8

I

installed modules, 10

O

old versions, 6

P

partial upgrade, 4

R

request prioritization, 6

S

stats, 6

T

technical process, 7

time estimation, 4

U

unique ID, 11

upgrade, 3

upgrade process, 3

upgrade scripts release time, 6