

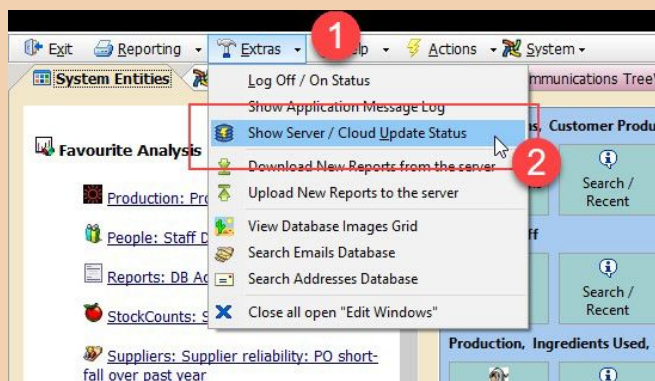
# Synchronizing Data between users in a Multi-server Orixo App

Orixo includes the capacity to be set up in multi-server mode with different users in multiple locations. Each location can house a copy of the database called an "instance". Users in each location can add, edit and view all the data. From time to time an update process can be run between database-instances, synchronizing the data.

This facility is useful if a business needs a system to work in multiple locations, for example with multiple warehouses, with some staff remote-working from home, or field-staff who are mobile. In multi-server mode Orixo holds data locally on each computer, so work can continue even when locations are not connected to the internet. In situations where the internet or data-network is weak and unreliable this can be very valuable. With a Multi-server system the business always has multiple copies of the data, in each separate instance.

A multi-server system is also safer, as damage to any one instance of the database still leaves the other instances in place. However, multi-server systems are more complicated, as having data in multiple locations means care needs to be taken with the logic of how data is managed to ensure updates by staff in one site do not overwrite updates from staff in another site. Orixo has well-developed systems to ensure that all of this is possible, and that (where necessary) change-logs of data are also kept so changes made by staff can be rolled back and reviewed.

## Updating Data between Instances



Accessing the Updates Screen

1. In the main menu of your system click "Extras."
2. In the menu that appears click "Show Server / Cloud Update Status."

**NOTE:** If your system is a "local" system with just one database, you will not see the "Show Server / Cloud Update Status" item in the menu. In this case your system does not need to be updated.

### Update Screen

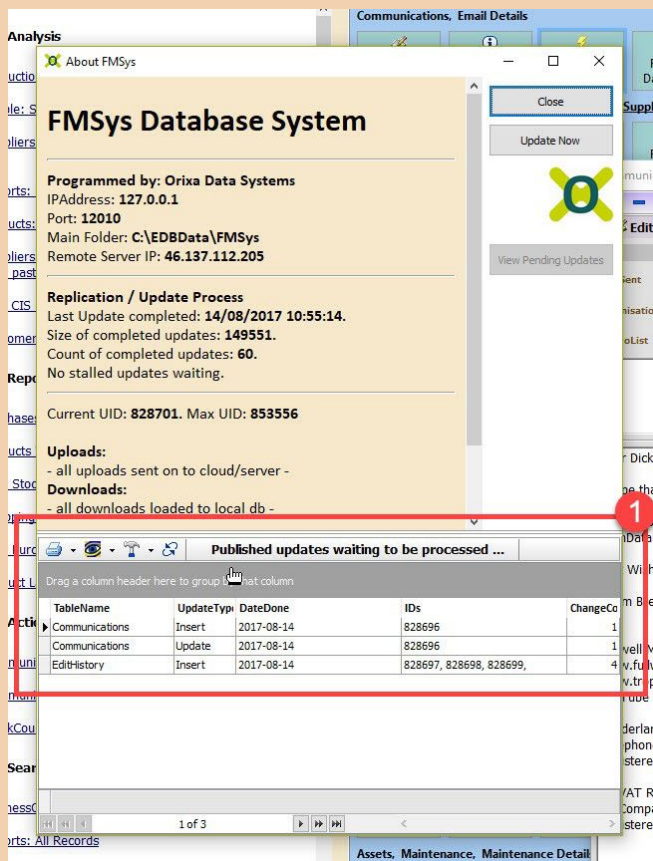
This screen includes data on the situation with your system, including the date of the last update, whether there are stalled updates and other system information.

It also shows the details of the information which is waiting to be updated (at 1. in the image), that is all the new or changed data that has been added since the last update.

To Run Updates, just click the "Update Now" button. This starts a database procedure "RunDataUpdates()", which is explained below.

The system will show messages in the message-log describing progress of the update process.

**NOTE:** You can double-click on the rows in the "updates waiting to be processed" grid to open a view of these records, so you can check over them if you want to.



Updates Screen

## Basic Concepts of remote-data with Orixia

1. Each instance stores a full copy of the main database, each instance can have multiple users who add their own data and this is stored locally. There is no need for a network or internet connection while using the system. The system is always fast and responsive because it is working with local data. Data is safely held on the instance and cannot be lost if a remote-server or internet connection fails.
2. Until each user clicks "Update Now" **only users connected to that instance** can see new data they have added since the previous update, and similarly they cannot see new data added by others since the last update.
3. As there may be many users adding data, and all sharing it at varied times, each user's version of the system will tend to be a bit different from others. However all data will be visible once updates are carried through across the whole system.

## Complex Issues

Orixia data is inter-linked in complex ways, with "parent", "child", "extension" and other complex data-types. Deletion of data such as Parent data can leave child data "orphaned". In Multi-site systems this process becomes more complex as it is potentially possible to delete "parent" records while other users add "child", creating orphans. If Multi-site system are to work well it is important that BusinessObjects are well designed to ensure that users cannot easily delete Parent records for which other users might be creating Children. Orixia contains code to cope with "orphan" data, but this code will result in users being confused as records they thought they had deleted will "reappear" because other users have added child-data to them on another database-instance.

## Understanding the Data-update process

The next section is technical, and probably not of interest to general users, but is included here so technical users can read it.

When users click "Update Now", a Database Procedure called "Updates\_Execute" is run. The details of the SQL for this procedure can be seen in the DB Management Utility so you can fully understand how it works. When this procedure is called a set of steps occur:

1. The instance calls "Save Updates" on itself, and its updates are saved locally to a file in the "Updates" database-store, with a file-name formed of the current date and time, plus the file-extension "EDBUpd."

2. This update file is copied across the network to the ServerUploads store, where the central server will use it later. This store is usually on a cloud-server. A network connection is required for this step. If no network connection is available the process stops, but the file is saved on disk and will transfer next time, possibly with other newer update-files. Note that travelling staff can "Save Updates" and then wait until they return to the office to actually transfer them to the ServerUploads store via a LAN network if this is easier.
3. The instance queries its remote "ServerDownloads" store to find the updates that are waiting to come to it from the server. A network connection is required for this step, again if none exists the process is passed over as in step 2.
4. If there are any updates waiting, these are copied across the network to the local store, if there is any issue with the network, non-transferred files are saved for use at a future time.
5. Then the instance calls "Load Updates" on each of the update files that have been transferred in order, adding their changes to the local database-instance.
6. All "used" update files are re-named with the extension "OLD" so they will not be re-used. They are not deleted immediately, but will be deleted after several days by the server. The "OLD" files are kept so changes through the update process can be tracked and reviewed if necessary.
7. The "Load Updates" process itself may cause changes in the instance, if any of these need to be propagated a further update file is created and also transferred to the "ServerDownloads" store.