

Synchronize Global Address Book to a contact folder (gab2contacts)

Gab2contacts is a small script that allows you to synchronize the Global Address Book (GAB) into a simple contacts folder (e.g. in the public folder).

This way your GAB can easily be synchronized to mobile devices (e.g. by sharing that public contacts folder globally).

It's designed to run regularly by using a cron or similar job depending on your preferences. It's lightweight and could run e.g. every hour.

It will detect changes, meaning that only contacts changed in the GAB are created, updated or deleted in the contacts folder. Items created there manually are not touched, even if the script is called with the "delete flag" to remove all previously created contacts.

If changes are made to a contact coming from the GAB, these changes are lost on the next update!

Configuring the gab2contact Script



The scripts and configuration files in the following article are part of the package **z-push-kopano-gab2contacts**

The gab2contact sync script can be run on any host that has the php-mapi module installed and that has access to your Zarafa/Kopano system. It has a dependency on z-push-common that needs to be installed as well. The script also brings its own config file (*config.php*), which needs to be customised to the local environment, so that the script can login at your Kopano/Zarafa system with admin privileges.

Excerpt of the gab2contact config.php. *SERVER*, *USERNAME* and *PASSWORD* may need to be adjusted to the local installation if the script is running on a different system than your Zarafa/Kopano installation. If Z-Push and Zarafa/Kopano are running on the same system no adjustments are necessary.



Multi-Tenant/Hosted Environments

At the moment this script is not multi-tenant compatible, as one contact folder store and folderid need to be filled it.



Multi-Server Environments

If you have several Kopano servers, you need to connect to the server/socket where your target store is located. Redirection (connect to any server to open a store) will not work.

If your ldap config varies or you want to use another backend to connect to, you have to adjust these values. Normally you can keep them as they are.

```
// The field to be hashed that is unique and never changes
// in the entire lifetime of the GAB entry.
define('HASHFIELD', 'account');

// ContactWorker implementation to be used
define('CONTACTWORKER', 'Kopano');
```

Setup the connection to the Kopano/ZCP system. The GAB source should be maintained.

```

// Server connection settings
// Depending on your setup, it might be advisable to change the lines below to one defined with your
// default socket location.
// Normally "default:" points to the default setting ("file:///var/run/kopano/server.sock")
// Examples: define("SERVER", "default:");
//           define("SERVER", "http://localhost:236/kopano");
//           define("SERVER", "https://localhost:237/kopano");
//           define("SERVER", "file:///var/run/kopano/server.sock");
// If you are using ZCP >= 7.2.0, set it to the zarafa location, e.g.
//           define("SERVER", "http://localhost:236/zarafa");
//           define("SERVER", "https://localhost:237/zarafa");
//           define("SERVER", "file:///var/run/zarafad/server.sock");
// For ZCP versions prior to 7.2.0 the socket location is different (http(s) sockets are the same):
//           define("SERVER", "file:///var/run/zarafa");

define('SERVER', 'default:');

define('USERNAME', 'SYSTEM');
define('PASSWORD', '');
define('CERTIFICATE', null);
define('CERTIFICATE_PASSWORD', null);

// The GAB to be used. This only needs to be set on a multi-tenant system.
// For standard installations, keep it at 'default'.
define('SOURCE_GAB', 'default');

```

Setup the store and the folderid (of a contacts folder) where the GAB should be synchronized to.

The script can sync the GAB only to one store/folder. If you want to give access to this folder to several people, use the sharing system. SYSTEM should be used for the public folder.

To find out the folderid use the kopano listfolders script located in `/usr/share/z-push/backend/kopano/listfolders.php`

```

// Store where the target contact folder is located.
// For the public folder, use SYSTEM.
// To use another store, use the same as USERNAME
// or another store where USERNAME has full access to.
define('CONTACT_FOLDERSTORE', 'SYSTEM');

// Set the target FolderId.
// You can find the id e.g. with the listfolders script of Kopano backend.
define('CONTACT_FOLDERID', '');

```

Runtime example of z-push-gab2contacts:

```

root@web:/etc/z-push# z-push-gab2contacts -a sync
Sync - Objects in GAB: 18 - in folder (before run): 0 - created: 16 - updated: 0 - deleted: 0 - untouched: 0
- ignored: 2

```

Executing it again just verifies if something changed and updates contacts as required:

```

root@web:/etc/z-push# z-push-gab2contacts -a sync
Sync - Objects in GAB: 18 - in folder (before run): 16 - created: 0 - updated: 0 - deleted: 0 - untouched:
16 - ignored: 2

```

Hidden GAB entries

If contacts or accounts are marked as hidden in the GAB they won't be visible in configured contact folder when using this script. This is done via the `kopanoHidden` or `zarafaHidden` flag (depending on your LDAP schema).

Versions prior to KC 8.2 require additional configuration so hidden entries are not synchronized. It's required to add the following line to your `/etc/kopano/ldap.propmap.cfg`

```
# PR_EC_AB_HIDDEN
0x67A7000B      = kopanoHidden
```

If you use the Zarafa LDAP schema you need do the same, but changing `/etc/zarafa/ldap.propmap.cfg` adding the mapping of the 'zarafaHidden' flag.

When using KC 8.2 or newer, check `/usr/share/kopano/ldap.propmap.cfg` if this already mapped.

After doing this change it's required to restart `kopano-server`.

Executing the `sync-gab` script will output an additional line for hidden users, like:

```
Kopano->GetGAB(): Ignoring user 'info-redirect' as account is hidden
```

Automating execution of `z-push-gab2contacts`

Depending on the amount of changes to the GAB of your Kopano system you should setup a cronjob to invoke `z-push-gab2contacts` from time to time. For normal system we recommend to add the following snippet to your daily cron run (e.g. `/etc/cron.daily`):

```
#!/bin/sh
set -e

GABCONTACTS=/usr/share/z-push/tools/gab2contacts/gab2contacts.php

if [ -e $GABCONTACTS ]; then
    $GABCONTACTS -a sync
fi
```

If the script is running on the same system as your Kopano, then you could also add the above script to `/etc/kopano/userscripts/createuser.d` to additionally run it whenever a new user has been created.