

Scripts & Stuff

- [Configuring Office 365 App Registration and Obtaining an Access Token for OneDrive on Linux](#)
- [Updating the OneDrive Client Service on Linux for Office 365 Integration](#)

Configuring Office 365 App Registration and Obtaining an Access Token for OneDrive on Linux

To allow the OneDrive client on Linux to access your Office 365 account, you need to create an Azure AD app registration in the Microsoft Azure portal. This app registration allows OneDrive to securely authenticate with your Office 365 account.

Prerequisites

1. **Microsoft Office 365 Account** with administrative privileges to create app registrations.
2. **Azure AD Access** for configuring app permissions.

Step 1: Create an App Registration in Azure AD

1. **Log in to Azure Portal:** Go to <https://portal.azure.com> and log in with your Office 365 credentials.
2. **Navigate to Azure Active Directory:** In the left sidebar, select **Azure Active Directory**.
3. **Go to App Registrations:** In the **Azure AD** dashboard, select **App registrations** from the sidebar.
4. **Create a New Registration:**
 - Click **New registration**.
 - **Name:** Enter a name, such as `OneDrive Linux Client`.
 - **Supported account types:** Select **Accounts in this organizational directory only**.

- **Redirect URI:** Choose **Web** and enter `http://localhost:53682/`. This is a required field for local testing, but the `onedrive` client will manage redirection internally.
 - Click **Register**.
5. **Copy the Application (Client) ID and Directory (Tenant) ID:** After registration, you'll see the **Overview** page with the **Application (client) ID** and **Directory (tenant) ID**. Copy these values to use later in the OneDrive client configuration.

Step 2: Configure API Permissions

1. **Go to API Permissions:** On your app's **Overview** page, select **API permissions** from the left sidebar.
2. **Add OneDrive API Permissions:**
 - Click **Add a permission**.
 - Select **Microsoft Graph**.
 - Choose **Delegated permissions**.
 - In the permissions search bar, type and select the following permissions:
 - `Files.Read`
 - `Files.ReadWrite`
 - `Files.Read.All`
 - `Files.ReadWrite.All`
 - `offline_access`
 - `User.Read`
 - Click **Add permissions**.
3. **Grant Admin Consent:** After adding the permissions, click **Grant admin consent for [Your Organization Name]**. This will authorize these permissions for all users in your organization. You may need to authenticate again to confirm the consent.

Step 3: Generate the Client Secret

1. **Go to Certificates & Secrets:** In your app's settings, select **Certificates & secrets** from the left sidebar.
2. **Create a New Client Secret:**
 - Under **Client secrets**, click **New client secret**.
 - **Description:** Enter a description, like `OneDrive Token`.
 - **Expires:** Choose the expiration duration (e.g., 6 months, 1 year, or 2 years). Note that you'll need to regenerate this token after it expires.
 - Click **Add**.
3. **Copy the Client Secret Value:** After saving, copy the **Value** of the client secret. This is your **Client Secret**, and you'll need it for the OneDrive client configuration. **Make sure to save it** securely, as it won't be displayed again.

Step 4: Obtain the Authorization Code

The OneDrive client needs an authorization code to generate an access token. To obtain this code:

1. **Construct the Authorization URL:** Replace `[TENANT_ID]`, `[CLIENT_ID]`, and `[REDIRECT_URI]` with your specific details in the URL below:

```
https://login.microsoftonline.com/[TENANT_ID]/oauth2/v2.0/authorize?client_id=[CLIENT_ID]&response_type=code&redirect_uri=[REDIRECT_URI]&response_mode=query&scope=offline_access%20Files.ReadWrite.All%20User.Read
```

- **[TENANT_ID]:** Your Directory (Tenant) ID from Step 1.
- **[CLIENT_ID]:** Your Application (Client) ID from Step 1.
- **[REDIRECT_URI]:** `http://localhost:53682/`.

Example URL (replace with actual IDs):

```
https://login.microsoftonline.com/12345678-1234-1234-1234-123456789abc/oauth2/v2.0/authorize?client_id=abcdefgh-ijkl-mnop-qrst-uvwxyz123456&response_type=code&redirect_uri=http://localhost:53682/&response_mode=query&scope=offline_access%20Files.ReadWrite.All%20User.Read
```

2. **Visit the Authorization URL:** Open this URL in your web browser. Sign in with your Office 365 account if prompted.
3. **Copy the Authorization Code:** After logging in, you'll be redirected to `http://localhost:53682/` with an authorization code in the URL. Copy the code from the URL; it will look something like this:

```
http://localhost:53682/?code=AUTHORIZATION_CODE
```

The `AUTHORIZATION_CODE` is what you need to proceed.

Step 5: Configure the OneDrive Client with OAuth Tokens

1. **Run the OneDrive Client:** Run the following command to configure the OneDrive client:

```
onedrive
```

This will start the interactive setup. If it detects that the client ID and secret are needed, it will prompt you for these.

2. **Provide Client ID, Secret, and Other Details:** Enter the following details when prompted:
 - **Client ID:** The Application (Client) ID you saved earlier.
 - **Client Secret:** The Client Secret you saved earlier.
 - **Tenant ID:** The Directory (Tenant) ID.
 - **Authorization Code:** Paste the authorization code you copied in Step 4.

3. **Test Syncing:** After configuration, test the OneDrive sync by running:

```
onedrive --synchronize
```

This command should successfully authenticate with Office 365 and start syncing your files.

Step 6: Configure the OneDrive Service to Use the New Tokens

To ensure that the OneDrive client uses the new tokens and starts automatically in monitor mode:

1. **Enable the OneDrive Service:** If the service is already set up as shown in the previous guide, simply start it:

```
systemctl --user start onedrive
```

2. **Enable Auto-Start:** If you haven't already, enable the OneDrive client to start on login:

```
systemctl --user enable onedrive
```

3. **Monitor Service Status:** Check the status of the OneDrive service to confirm it's running properly.

```
systemctl --user status onedrive
```

Step 7: Troubleshooting Tips

- **Invalid Token:** If the token expires, repeat the above process to generate a new authorization code.
- **Permissions Error:** Ensure that permissions were granted in Step 2. Check Azure AD's **API Permissions** to verify.
- **Service Fails to Start:** Review logs with `journalctl --user -u onedrive.service -f` to see any specific errors.

Summary

This guide provides a step-by-step solution for configuring Azure AD app registration, obtaining OAuth tokens, and setting up the OneDrive client on Linux to sync with Office 365. By following these steps, you can create a seamless integration between your Linux system and Office 365's OneDrive, enabling continuous file synchronization.

Updating the OneDrive Client Service on Linux for Office 365 Integration

This guide provides detailed steps for updating the `onedrive` client on a Linux system. This client allows seamless integration between Linux and Microsoft Office 365, providing continuous synchronization of your OneDrive files. By following this guide, you'll learn how to update the OneDrive client and manage it as a systemd user service that runs in monitor mode.

Prerequisites

1. **Existing OneDrive Client Installation:** This guide assumes that the `onedrive` client is already installed on your system.
2. **D Compiler:** The OneDrive client requires a D compiler, such as `ldc2`.
3. **Admin (sudo) Access:** You'll need sudo privileges to install or update system-wide software.
4. **Internet Connection:** Required for downloading and installing updates.

Step 1: Verify the Current Version of OneDrive

Before updating, check the version of your currently installed OneDrive client. Run:

```
onedrive --version
```

If `onedrive` is located in `/usr/local/bin`, use the full path to verify:

```
/usr/local/bin/onedrive --version
```

Note the current version number, as it will help confirm that the update was successful.

Step 2: Stop the Existing OneDrive Service

If the OneDrive client is running as a systemd user service, stop it before updating to prevent conflicts.

1. **Check the Service Status:**

```
systemctl --user status onedrive
```

2. **Stop the Service:**

```
systemctl --user stop onedrive
```

3. **Disable the Service Temporarily:** This prevents the service from starting automatically during the update process.

```
systemctl --user disable onedrive
```

Step 3: Download and Install the Latest Version of the OneDrive Client

1. **Remove the Existing Binary:** To avoid conflicts, remove the old version of `onedrive` from `/usr/local/bin` or `/usr/bin`:

```
sudo rm /usr/local/bin/onedrive
```

2. **Install Prerequisites:** Install required dependencies, including the D compiler and development libraries.

```
sudo apt update  
sudo apt install -y libcurl4-openssl-dev libsqlite3-dev libnotify-dev ldc
```

3. **Download the Latest Version:** Clone the official OneDrive client repository from GitHub.


```
git clone https://github.com/abraunegg/onedrive.git
cd onedrive
```

4. **Build and Install:** Configure, compile, and install the client. This step may take a few minutes.

```
./configure
make
sudo make install
```

5. **Verify the New Version:** Check that the updated version is installed correctly.

```
onedrive --version
```

Step 4: Configure the OneDrive Systemd User Service

To run `onedrive` in monitor mode continuously, we'll set up a systemd user service. This ensures that `onedrive` starts automatically in the background and restarts if it stops.

1. **Create or Edit the Systemd Service File:** Place the service file in `~/.config/systemd/user/onedrive.service`. This ensures it runs as a user service without requiring root privileges.

```
mkdir -p ~/.config/systemd/user
nano ~/.config/systemd/user/onedrive.service
```

Paste the following configuration:

```
[Unit]
Description=OneDrive Free Client
Documentation=https://github.com/abraunegg/onedrive
After=network-online.target

[Service]
ExecStart=/usr/local/bin/onedrive --monitor
Restart=on-failure
RestartSec=3
```

```
[Install]
```

```
WantedBy=default.target
```

- **ExecStart:** Runs `onedrive` in `--monitor` mode to continuously sync changes.
 - **Restart:** Ensures the service restarts if it fails.
 - **WantedBy:** Sets the service to start automatically for the current user.
2. **Reload the Systemd Daemon:** After modifying service files, reload the systemd daemon to recognize changes.

```
systemctl --user daemon-reload
```

3. **Enable and Start the Service:** Enable and start the `onedrive` service to run it in the background.

```
systemctl --user enable onedrive  
systemctl --user start onedrive
```

4. **Check the Service Status:** Verify that the service is running in monitor mode.

```
systemctl --user status onedrive
```

The status output should indicate that `onedrive` is active and running in monitor mode. You'll see real-time logging output from the OneDrive client.

Step 5: Test the Updated OneDrive Client

To confirm that the updated client is working as expected:

1. **Manual Sync Test:** You can run a manual sync to check for any synchronization issues.

```
onedrive --synchronize
```

2. **Automatic Monitoring:** The `--monitor` mode will automatically sync changes between your OneDrive cloud and your local folder. Test this by creating, modifying, or deleting files in your OneDrive folder and confirming that the changes sync correctly.
3. **View Logs:** To view logs of recent sync events, use the following command:

```
journalctl --user -u onedrive.service -f
```

This command will show real-time logs of OneDrive activity.

Troubleshooting Common Issues

- **Service Not Starting:** Ensure that the service file path is correct and that `ExecStart` points to the correct `onedrive` binary location.
- **Permission Denied Errors:** If permissions are causing issues, ensure that you have the correct ownership and permissions for the OneDrive sync folder.
- **No Network Connection:** Ensure that `onedrive` is set to start after the network is online by including `After=network-online.target` in the service file.

Updating in the Future

For future updates, simply repeat **Step 3** to download and install the latest version. You won't need to reconfigure the systemd service unless there are changes to the service file requirements.

Summary

This guide provides a complete solution for setting up and updating the `onedrive` client on Linux to sync with Office 365. With this setup:

- The OneDrive client runs continuously in monitor mode.
- It synchronizes automatically on boot.
- You can manage the service with `systemctl --user` commands.

This configuration provides a reliable and seamless experience for Linux users integrating with Microsoft Office 365's OneDrive, allowing files to sync effortlessly in the background.