

Quick tutorial Google Sites and Google Project hosting

Javier Quevedo Fernández

What do I need?	3
Google Sites	3
<i>Create a new site</i>	3
<i>Modify the content</i>	4
<i>Invite collaborators</i>	4
Google Code Hosting	5
<i>Bug tracking system</i>	5
<i>Create a project</i>	5
<i>Invite collaborators</i>	6
<i>Checkout a local copy of the respository</i>	6
<i>Creating files and committing changes</i>	7
<i>Update an outdated copy</i>	8
<i>Resolving conflicts</i>	8

What do I need?

You only need a Google Account and a computer with a web browser and internet connection. If you do not have a Google Account, you can create one at <https://www.google.com/accounts/>

Google Sites

Google Sites is a free and easy to use **webpage / wiki hosting** tool. With Google Sites you can create pages and edit them directly in-site by one or multiple users and make them public or private.

Create a new site

1. Browse <http://sites.google.com/>
2. Click on *Create New Site*
3. Give it a name, a description and choose your favorite theme.
4. Done!

The result should look something like the following picture.



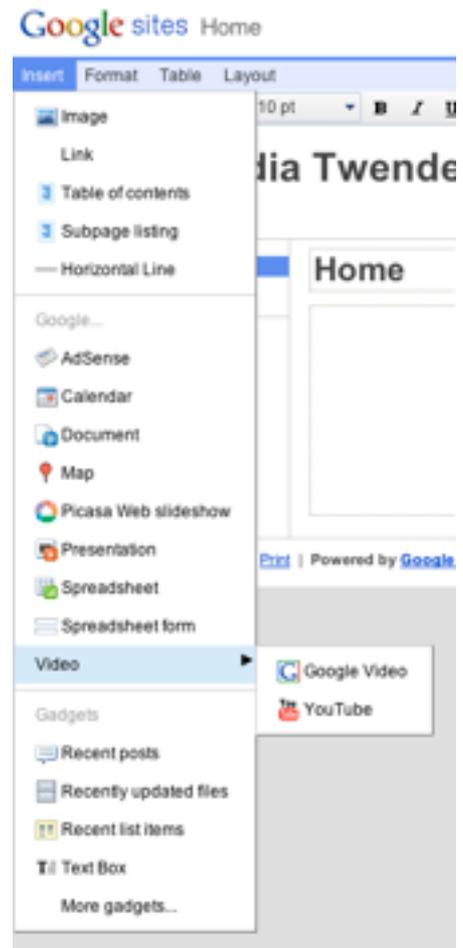
Modify the content

1. Click on Edit page
2. Make your changes
3. Add videos, images, documents or links to other pages with the Insert menu
4. Additionally, the style and layout of the document can be modified using the Layout and Format menu items
5. Save it

You changes are immediately available and can be browsed at the correspondent URL.

Invite collaborators

1. Click on *More Actions*
2. Click on *Share this site*
3. Enter the email addresses and roles of the desired collaborators.
4. Send the invitation
5. If the receiver accepts, he/she is now an editor as well.



Invite people to your site <input type="radio"/> as owners <input checked="" type="radio"/> as collaborators <input type="radio"/> as viewers <input type="text"/> <small>Separate email addresses with commas</small> Choose from contacts <input type="button" value="Invite these people"/>	People with access Owners (1)  Remove Collaborators (0) Viewers (0)
Advanced permissions <input checked="" type="checkbox"/> Anyone in the world may view this site (make it public)	
Sharing with coworkers? Companies can securely manage multiple sites. Learn more	

Google Code Hosting

Google code is free a project management tool. Each project includes a source code repository, a Wiki, and a bug tracking system.

Bug tracking system

Create a new issue by clicking on Issues -> New Issue

Enter a description of the issue, you may upload files and add labels to the issue. Submit it.

Create a project

1. Go to <http://code.google.com/hosting/createProject>
2. Set the project information.
3. Under *version control system* select *Subversion*.
4. Click on create



The screenshot shows the Google Code project page for 'testprojecttwende'. At the top left is the project logo, a computer monitor with a green 'G' on it. To its right is the project name 'testprojecttwende' and a search box with the text 'Search projects'. Below the logo and name are navigation tabs: 'Project Home', 'Downloads', 'Wiki', 'Issues', 'Source', and 'Administer'. Under 'Project Home' are links for 'Summary', 'Updates', and 'People'. A green tip box says: 'Tip: Project owners, see our [Getting Started](#) guide for steps to configure your project. [hide](#)'. The main content area has a light blue background and contains several sections: 'Star this project' with a star icon, 'Code license:' with a blurred link, 'Labels:' with a blurred link, 'Feeds: [Project feeds](#)', and 'Project owners:' with a blurred link and a 'People details' link. At the bottom, there is a copyright notice: '©2009 Google - [Code Home](#) - [Terms of Service](#) - [Privacy Policy](#) - [Site Directory](#) - [Project Hosting Help](#)'. Below this is the text 'Hosted by 

Invite collaborators

Click on Administer -> Project Members, and enter the email addresses of the project owners, committers and contributors.

Checkout a local copy of the repository

1. Download and install Tortoise svn from <http://tortoisesvn.net/downloads>
2. Create a local folder
3. Copy the projects url

Command-Line Access

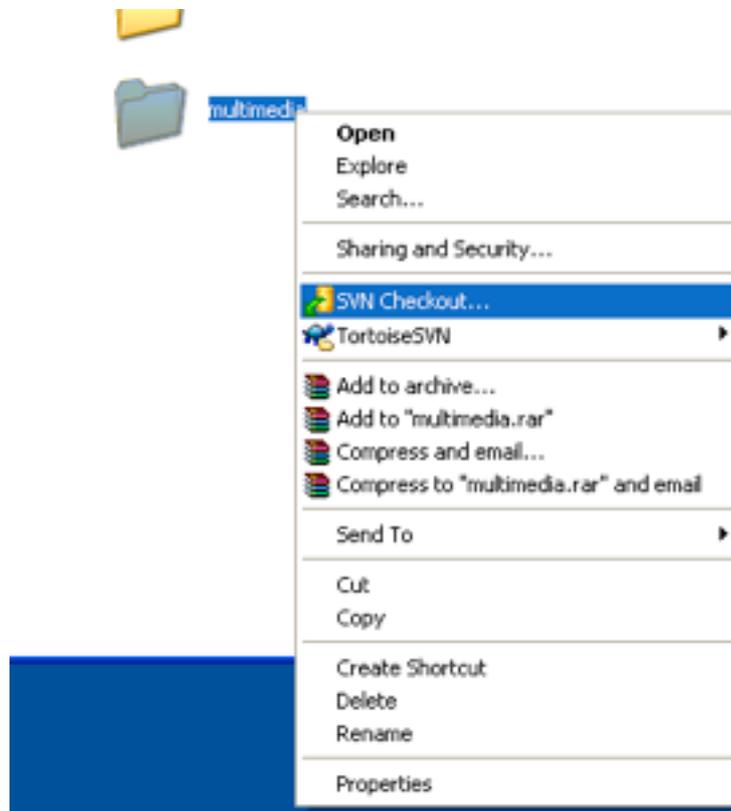
If you plan to make changes, use this command to check out the code as yourself using HTTPS:

```
# Project members authenticate over HTTPS to allow committing changes.  
svn checkout https://testprojecttwende.googlecode.com/svn/trunk/ testpro  
  
When prompted, enter your generated googlecode.com password.
```

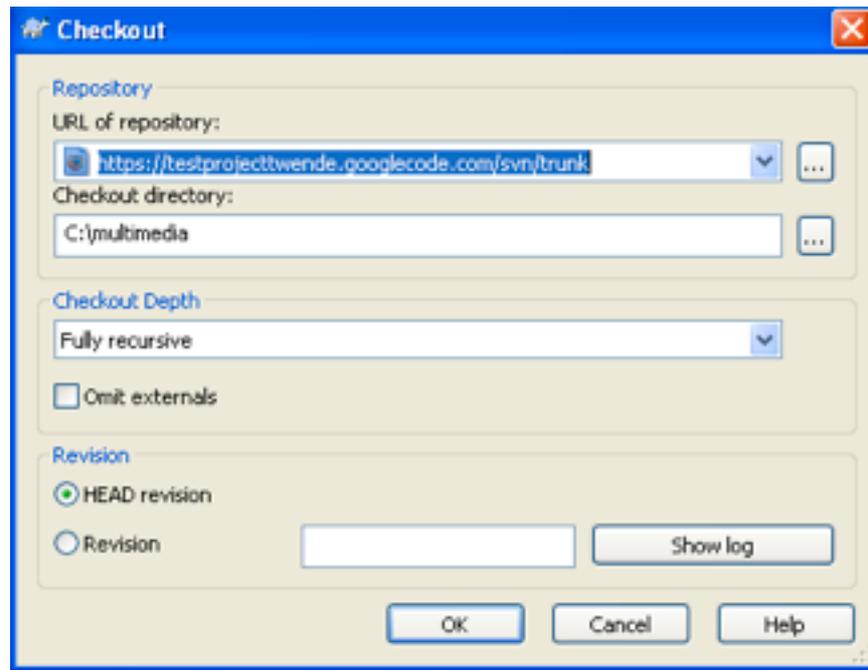
Use this command to anonymously check out the latest project source code:

```
# Non-members may check out a read-only working copy anonymously over HT  
svn checkout http://testprojecttwende.googlecode.com/svn/trunk/ testproj
```

4. Right click on the Folder and select SVN Checkout...



5. Enter the URL

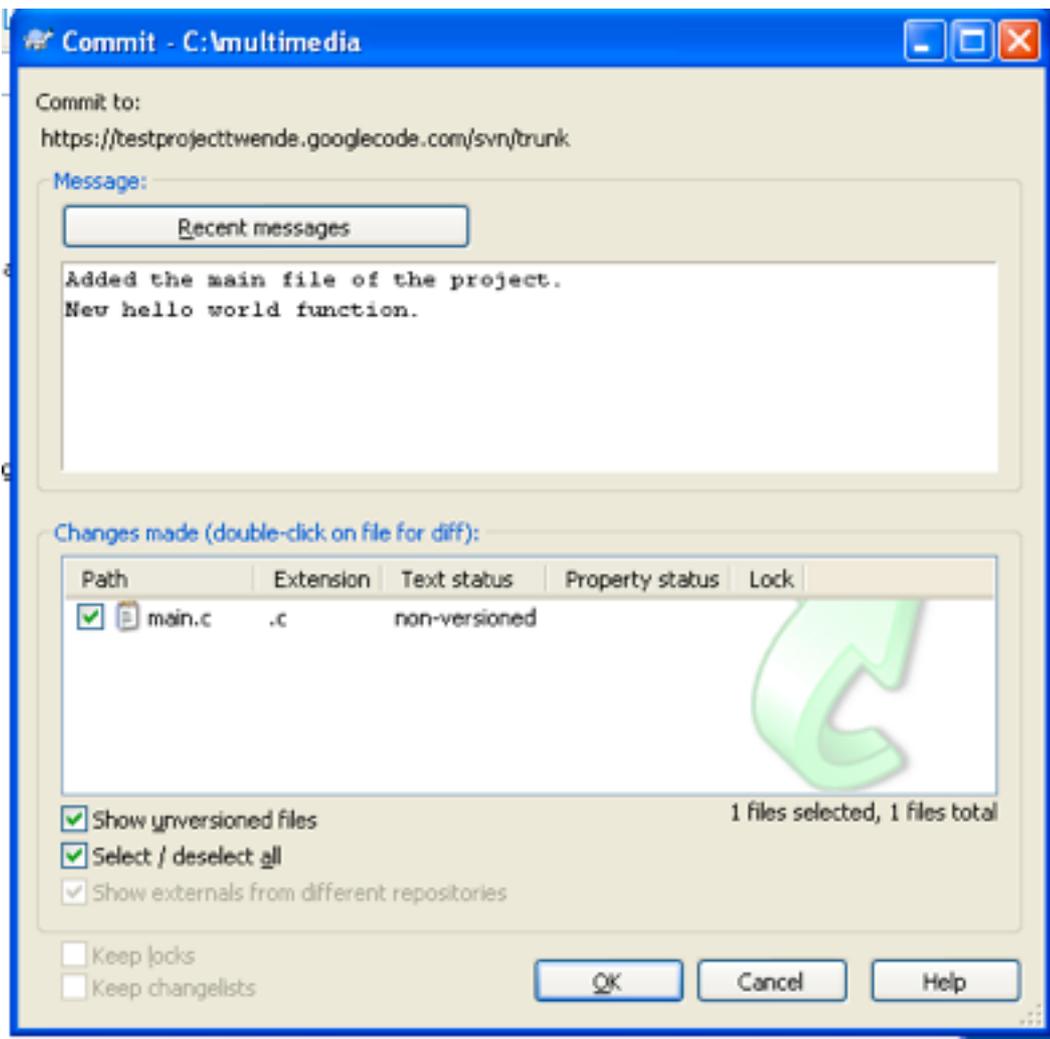


6. When prompted for credential, enter the *username* and the *Generated password*

7. Done!

Creating files and committing changes

1. Create a new file in the folder, for example *main.c*
2. Make changes and save the file.
3. Right click on the Projects folder, select *SVN Commit*.
4. Select the files that you want to *add* to the repository
5. Enter a textual message with the changes done



6. Select OK
7. The changes will be uploaded to the repository. Done.

Update an outdated copy

In order to update a local copy right click on the projects folder and select SVN Update.

The old file will be updated and the new ones created.

Resolving conflicts

If more than one person are committing and updating changes at the same time, at a certain moment conflicts will appear due to the concurrent modification of the same file. These conflicts have to be solved by the user, but Subversion will give us a hand.

Subversion will create three files. A file *main.mine* which is a copy of the file I was trying to commit. It will create an additional file *main.c.r09* with the oldest version out of conflict and a *main.c.r10* with the version that we are having conflict with. The file *main.c* will remain but it will include a merge of both files

with comments determining which part is common, and which part belongs to which version.

In order to solve the conflict we must set the correct changes without conflict in the file *main.c* and delete all of the other versions. Now we are ready to commit the current version as the one without conflict.