

Kollaborate Server



Scripting Guide (Beta 1)

Kollaborate Server can run local scripts that respond to events like file uploading or user account creation, giving you the power to customize Kollaborate Server and fully integrate it with other in-house systems and tools.

Please note that this is a beta and the scripting syntax may change before the final release.

Requirements

- Kollaborate Server
- OS X, Linux or Windows computer
- [Python 2.7 \(not 3.x\)](#)

Contents

Prerequisites	3
Installing	4
Scripting Events	5
File Status Changed	5
File Uploaded	6
File Deleted	6
File Encoded	7
File Encoding Failed	8
File Viewed	8
Comment Created	9
Comment Changed	10
Comment Deleted	11
Project Created	11
Project Changed	12
Project Deleted	13
User Created	14
User Added to Project	14
Task Created	15
Task Changed	16
Task Deleted	17
Share Link Created	18
Share Link Changed	18
Share Link Deleted	19
Share Link Viewed	20
Appendix A: Using Different Scripting Languages	22

Prerequisites

This guide assumes you have already installed and configured Kollaborate Server (see the separate **Kollaborate Server Installation Guide** manual if not) and that you have basic experience with Python and an understanding of Python code syntax.

Important note: Python is a language where indentation and white space is important. Thus it is easy to introduce indentation errors into your code when copying and pasting from the web or this user manual. We therefore recommend you delete the spacing in front of every line and re-indent after pasting it in.

Installing

Scripting is built into Kollaborate Server 1.3.1 Scripting Beta and higher.

1. Follow the installation steps in the **Kollaborate Server Installation Guide** manual as normal.
2. Make sure Python 2 is installed on your system:

Ubuntu - Type **sudo apt-get install python** at a command prompt.

Mac - Python 2 should be already installed, which can be verified by typing **which python** into a Terminal prompt. If that command returns no results, please [download Python](#) directly.

Windows - Python is not installed by default and can be downloaded from [here](#).

3. After logging into the Admin Area of Kollaborate Server, click the Configure page.
4. Under Scripting, set **Python Path** to the path of your Python executable. You can find this out by typing **which python** into a command prompt on Mac / Linux or **where python** on Windows.
5. Copy the **kollab_script.py** sample script from the **Sample Scripts** folder in the Kollaborate Server installation zip file to the parent folder of your web directory. For example, if your web directory is `/var/www`, you should copy it to the `/var` folder.

The sample script outputs data to the PHP log so you can see that the script's functions are being executed but does not perform any additional actions.

Scripting Events

The following events are available in custom scripts:

File Status Changed

Occurs when a file's status is changed, including when it is approved or rejected.

Syntax

```
def FileStatusChanged(file,project,user):
```

Parameters

The **file** variable has the following keys:

id - Numeric file identifier
title - File title
status - Numeric status ID.

Possible values are:

0 - No status
1 - Work in Progress
2 - Needs Approval
3 - Rejected
4 - Approved

The **project** variable has the following keys:

id - Numeric project identifier (-1 if it's the user's private storage area)
title - Project title

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task
name - User's first and last name

Return Value

None.

File Uploaded

Occurs when a file is successfully uploaded.

Syntax

```
def FileUploaded(file,project,user):
```

Parameters

The **file** variable has the following keys:

- id** - Numeric file identifier
- title** - File title
- path** - The full path to the file on disk

The **project** variable has the following keys:

- id** - Numeric project identifier (-1 if it's the user's private storage area)
- title** - Project title

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

File Deleted

Occurs when a file is deleted from a project.

Syntax

```
def FileDeleted(file,project,user):
```

Parameters

The **file** variable has the following keys:

- id** - Numeric file identifier

title - File title

path - The previous path to the file on disk

The **project** variable has the following keys:

id - Numeric project identifier (-1 if it's the user's private storage area)

title - Project title

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task

name - User's first and last name

Return Value

None.

File Encoded

Occurs when a file is successfully converted by Kollaborate Encoder. This event does not fire when uploading a file via an app or if Kollaborate Encoder is not setup.

Syntax

```
def FileEncoded(file,project,user):
```

Parameters

The **file** variable has the following keys:

id - Numeric file identifier

title - File title

path - The path to the encoded file on disk

The **project** variable has the following keys:

id - Numeric project identifier (-1 if it's the user's private storage area)

title - Project title

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task

name - User's first and last name

Return Value

None.

File Encoding Failed

Occurs when a file fails to be converted by Kollaborate Encoder. This event does not fire when uploading a file via an app or if Kollaborate Encoder is not setup.

Syntax

```
def FileEncodingFailed(file,project,user):
```

Parameters

The **file** variable has the following keys:

- id** - Numeric file identifier
- title** - File title
- path** - The path to the encoded file on disk

The **project** variable has the following keys:

- id** - Numeric project identifier (-1 if it's the user's private storage area)
- title** - Project title

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

File Viewed

Occurs every time a file is viewed.

Syntax

```
def FileViewed(file,project,user):
```

Parameters

The **file** variable has the following keys:

- id** - Numeric file identifier
- title** - File title
- path** - The path to the file on disk

The **project** variable has the following keys:

- id** - Numeric project identifier (-1 if it's the user's private storage area)
- title** - Project title

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Comment Created

Occurs when a new comment is added to a file.

Syntax

```
def CommentCreated(comment, file, project, user):
```

Parameters

The **comment** variable has the following keys:

- id** - Numeric comment identifier
- body** - Comment body
- timecode** - The timecode position of the comment (if a video or audio file)

The **file** variable has the following keys:

- id** - Numeric file identifier
- title** - File title

The **project** variable has the following keys:

id - Numeric project identifier (-1 if it's the user's private storage area)
title - Project title

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task
name - User's first and last name

Return Value

None.

Comment Changed

Occurs when a comment is edited.

Syntax

```
def CommentChanged(comment, file, project, user):
```

Parameters

The **comment** variable has the following keys:

id - Numeric comment identifier
body - Comment body
timecode - The timecode position of the comment (if a video or audio file)

The **file** variable has the following keys:

id - Numeric file identifier
title - File title

The **project** variable has the following keys:

id - Numeric project identifier (-1 if it's the user's private storage area)
title - Project title

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task
name - User's first and last name

Return Value

None.

Comment Deleted

Occurs when a comment is deleted from a file.

Syntax

```
def CommentDeleted(comment, file, project, user):
```

Parameters

The **comment** variable has the following keys:

- id** - Numeric comment identifier
- body** - Comment body
- timecode** - The timecode position of the comment (if a video or audio file)

The **file** variable has the following keys:

- id** - Numeric file identifier
- title** - File title

The **project** variable has the following keys:

- id** - Numeric project identifier (-1 if it's the user's private storage area)
- title** - Project title

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Project Created

Occurs when a new project is created.

Syntax

```
def ProjectCreated(project,user):
```

Parameters

The **project** variable has the following keys:

- id** - Numeric project identifier
- title** - Project title
- status** - Numeric field denoting the project status.

Possible values:

- 0 - Pre-Production
- 1 - Production
- 2 - Post Production

client - The name of the client, if entered

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Project Changed

Occurs when a project is edited.

Syntax

```
def ProjectChanged(project,user):
```

Parameters

The **project** variable has the following keys:

- id** - Numeric project identifier
- title** - Project title
- status** - Numeric field denoting the project status.

Possible values:
0 - Pre-Production
1 - Production
2 - Post Production

client - The name of the client, if entered

archived - Numeric field indicating if the project has been archived (0 = unarchived)

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task

name - User's first and last name

Return Value

None.

Project Deleted

Occurs when a project is deleted.

Syntax

```
def ProjectDeleted(project,user) :
```

Parameters

The **project** variable has the following keys:

id - Numeric project identifier

title - Project title

status - Numeric field denoting the project status.

Possible values:
0 - Pre-Production
1 - Production
2 - Post Production

client - The name of the client, if entered

archived - Numeric field indicating if the project has been archived (0 = unarchived)

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task

name - User's first and last name

Return Value

None.

User Created

Occurs when a new user account is created.

Syntax

```
def UserCreated(user):
```

Parameters

The **user** variable has the following keys:

- id** - Numeric ID of the added user
- name** - User's first and last name
- email** - User's email address

Return Value

None.

User Added to Project

Occurs when a user account is added to a project.

Syntax

```
def UserAddedToProject(user, project):
```

Parameters

The **user** variable has the following keys:

- id** - Numeric ID of the added user
- name** - User's first and last name
- email** - User's email address
- department_id** - Numeric ID of the department the user was added to

department_title - Title of the department the user was added to
position_id - Numeric ID of the position the user was assigned
position_title - Title of the position the user was assigned

The **project** variable has the following keys:

id - Numeric ID of the project
title - Title of the project

Return Value

None.

Task Created

Occurs when a new task is created.

Syntax

```
def TaskCreated(task,project,user) :
```

Parameters

The **task** variable has the following keys:

id - Numeric ID of the new task
title - Title of the task
description - Task description
due - Numeric timestamp of the due date based on the number of seconds since Jan 1st 1970
assigned_id - Numeric ID of the assigned user (-1 if unassigned)
assigned_name - Name of the assigned user
status - Numeric task status.

Possible values:

- 0 - Pending
- 1 - In Progress
- 2 - Needs Approval
- 3 - Completed

The **project** variable has the following keys:

id - Numeric ID of the project
title - Title of the project

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task
name - User's first and last name

Return Value

None.

Task Changed

Occurs when a task is edited.

Syntax

```
def TaskChanged(task, project, user) :
```

Parameters

The **task** variable has the following keys:

id - Numeric ID of the new task
title - Title of the task
description - Task description
due - Numeric timestamp of the due date based on the number of seconds since Jan 1st 1970
assigned_id - Numeric ID of the assigned user (-1 if unassigned)
assigned_name - Name of the assigned user
status - Numeric task status.

Possible values:

- 0 - Pending
- 1 - In Progress
- 2 - Needs Approval
- 3 - Completed

The **project** variable has the following keys:

id - Numeric ID of the project
title - Title of the project

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task
name - User's first and last name

Return Value

None.

Task Deleted

Occurs when a task is deleted.

Syntax

```
def TaskDeleted(task,project,user):
```

Parameters

The **task** variable has the following keys:

id - Numeric ID of the new task

title - Title of the task

description - Task description

due - Numeric timestamp of the due date based on the number of seconds since Jan 1st 1970

assigned_id - Numeric ID of the assigned user (-1 if unassigned)

assigned_name - Name of the assigned user

status - Numeric task status.

Possible values:

0 - Pending

1 - In Progress

2 - Needs Approval

3 - Completed

The **project** variable has the following keys:

id - Numeric ID of the project

title - Title of the project

The **user** variable has the following keys:

id - Numeric ID of the user who performed the task

name - User's first and last name

Return Value

None.

Share Link Created

Occurs when a new shared link is created.

Syntax

```
def ShareLinkCreated(link,file,project,user):
```

Parameters

The **link** variable has the following keys:

- id** - String ID of the link
- name** - Descriptive name of the link
- url** - Link URL
- password** - Password (if assigned)
- requires_password** - Whether a password is required (0 = not required)
- expiration** - Numeric expiration timestamp with the number of seconds since Jan 1st 1970. 0 if it does not expire.

The **file** variable has the following keys:

- id** - Numeric ID of the file
- title** - Title of the file

The **project** variable has the following keys:

- id** - Numeric ID of the project
- title** - Title of the project

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Share Link Changed

Occurs when a shared link is modified.

Syntax

```
def ShareLinkChanged(link,file,project,user):
```

Parameters

The **link** variable has the following keys:

- id** - String ID of the link
- name** - Descriptive name of the link
- url** - Link URL
- password** - Password (if assigned)
- requires_password** - Whether a password is required (0 = not required)
- expiration** - Numeric expiration timestamp with the number of seconds since Jan 1st 1970. 0 if it does not expire.

The **file** variable has the following keys:

- id** - Numeric ID of the file
- title** - Title of the file

The **project** variable has the following keys:

- id** - Numeric ID of the project
- title** - Title of the project

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Share Link Deleted

Occurs when a shared link is deleted.

Syntax

```
def ShareLinkDeleted(link,file,project,user):
```

Parameters

The **link** variable has the following keys:

- id** - String ID of the link
- name** - Descriptive name of the link
- url** - Link URL
- password** - Password (if assigned)
- requires_password** - Whether a password is required (0 = not required)
- expiration** - Numeric expiration timestamp with the number of seconds since Jan 1st 1970. 0 if it does not expire.

The **file** variable has the following keys:

- id** - Numeric ID of the file
- title** - Title of the file

The **project** variable has the following keys:

- id** - Numeric ID of the project
- title** - Title of the project

The **user** variable has the following keys:

- id** - Numeric ID of the user who performed the task
- name** - User's first and last name

Return Value

None.

Share Link Viewed

Occurs every time a user clicks or accesses a shared link URL.

Syntax

```
def ShareLinkViewed(link,file,project):
```

Parameters

The **link** variable has the following keys:

id - String ID of the link
name - Descriptive name of the link
url - Link URL

The **file** variable has the following keys:

id - Numeric ID of the file
title - Title of the file

The **project** variable has the following keys:

id - Numeric ID of the project
title - Title of the project

Return Value

None.

Appendix A: Using Different Scripting Languages

While Kollaborate Server uses Python for scripting, you can use it in conjunction with any programming language or software tool that is capable of being executed from the command-line.

To do this, use the Popen command in Python to launch another process on the system.

As an example, here is some Python code that calls some PHP code to send an email alert:

Syntax

```
class KollabScript:
    @staticmethod
    def FileStatusChanged(file,project,user):
        from subprocess import Popen, PIPE

        p = Popen(["php","-r", "mail('name@example.com','Kollaborate file
status changed','');"])
```