



Platform 201

We will record today's presentation and provide the recording and slides after.

Introduce yourself in the Chat!

Name

Pronouns

Organization

What brought you here today?





Training Norms

1. Please keep yourself muted, help us reduce background noise.
2. Raise your hand or type questions into the chat.
3. Feel free to ask other participants questions in the chat!
4. We all have varying levels of Platform knowledge on this call. Ask the questions you have—no question is too simple or too advanced!



Session 2

Scripts - SQL & Python

Version Control

Workflows

Questions



Scripts - SQL & Python

Creating Scripts

Parameters

Automation

Notifications

Scripts

Creating a new Script

Use the Code tab to access your Scripts

- To create a new Script click Code from the top nav bar then select the language you want to use under Scripts
 - For SQL Scripts the current database selected from your dropdown will be used
- Enter your code into the editor
- Click Run
- SQL Scripts that end in a SELECT statement produce a CSV file of the results.

The screenshot shows a web application interface with a top navigation bar containing 'QUERY', 'DATA', 'CODE', 'TOOLS', and 'PUBLISH'. The 'CODE' tab is active. Below the navigation bar, there are two tabs: 'Dashboard' and 'Activity'. The 'Activity' tab is selected, displaying 'Activity Metrics' for the last 30 days. The metrics are: 0 Scheduled, 10 Needs Attention, 0 In Progress, and 11 Succeeded. Below the metrics is a 'Favorites' table with columns for Type, Name, Owner, and Updated At. The table lists several items, including 'City of Chicago Food Inspection Dashboard', 'Script from Query', 'SQL: Select from Police Stations', 'IMPORT, CLEAN, MATCH, AND EXPORT DA', and 'Workflow Demo'. At the bottom of the page, there are sections for 'Organization Details' and 'Quick Links'.

| Type | Name | Owner | Updated At |
|-------------------------------------|---|------------|------------------------|
| <input checked="" type="checkbox"/> | City of Chicago Food Inspection Dashboard | - | 04/27/2022 2:51:30 PM |
| <input checked="" type="checkbox"/> | Script from Query | Jane Smith | 06/21/2023 11:16:26 AM |
| <input checked="" type="checkbox"/> | SQL: Select from Police Stations | Jane Smith | 08/24/2022 12:21:42 PM |
| <input checked="" type="checkbox"/> | IMPORT, CLEAN, MATCH, AND EXPORT DA | Jane Smith | 01/16/2024 7:03:39 PM |
| <input checked="" type="checkbox"/> | Workflow Demo | Jane Smith | 12/12/2023 10:17:42 AM |



Scripts

Parameters

- Platform allows you to use custom parameters in scripts
- These parameters can be set and accessed in the UI
- This feature allows more technical team members to write the code while less technical team members edit and run the scripts

Search

DATA CODE TOOLS PUBLISH Civis Database Admin JB

PROTOCOL *

Select...

Specify the transfer protocol to use. The options are 'FTP', 'FTP_TLS', or 'SFTP'.

HOST *

Enter the URL of the S/FTP location, e.g. 'ftp.examplesite.com'.

CREDENTIAL

Select...

[Create Credential](#)

Select the credential containing the appropriate username and password for the host.

Credential Password is a Private Key?

If the password to your FTP credential is a private key, check this box. In most cases it does not need to be checked.

PORT (ADVANCED)

Specify the port on which to establish the connection. If left blank the default is 21 for FTP and 22 for SFTP.

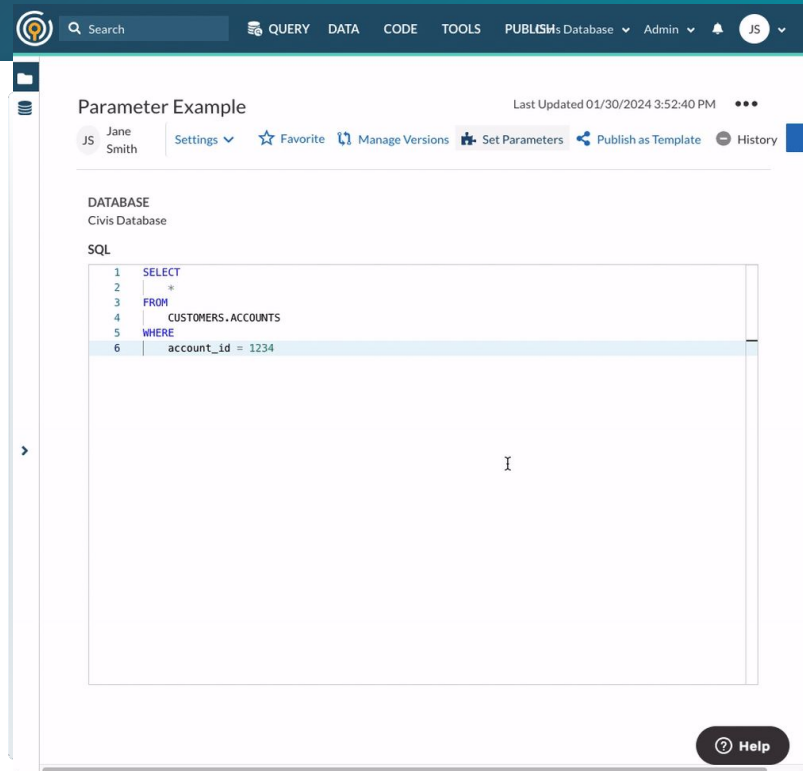
HOST DIRECTORY

Specify the directory within the host from which to import files, e.g. 'download/'.

Scripts

Parameters

- From any Platform job, click the puzzle icon “Set Parameter”. Within the Parameters menu:
 - Click “+ New Parameter”
 - Fill out the required fields, and optional fields as desired
 - Select “Add”
 - Select “Save” before
- Once you add a parameter, you should see it as an option in the script



The screenshot shows a web application interface for editing a script. At the top, there is a search bar and navigation tabs for QUERY, DATA, CODE, TOOLS, PUBLISH, Admin, and JS. The main content area is titled "Parameter Example" and shows a SQL script editor. The script is as follows:

```
1 SELECT
2     *
3 FROM
4     CUSTOMERS.ACCOUNTS
5 WHERE
6     account_id = 1234
```

The interface includes a "Set Parameters" button in the top right of the editor area. A "Help" button is located at the bottom right of the editor area. The script is currently empty, and the cursor is positioned at the end of the last line.

Scripts

Parameters

| Language | Parameter Format |
|-----------|--|
| SQL | Wrap parameters in double curly braces (e.g. <code>{{myParamName.literal}}</code>) |
| Python | Read parameters as environment variables (e.g. <code>os.environ['myParamName']</code>) |
| R | Read parameters as environment variables (e.g. <code>Sys.getenv('myParamName')</code>) |
| Container | Read parameters as environment variables formatted in the script's coding language |

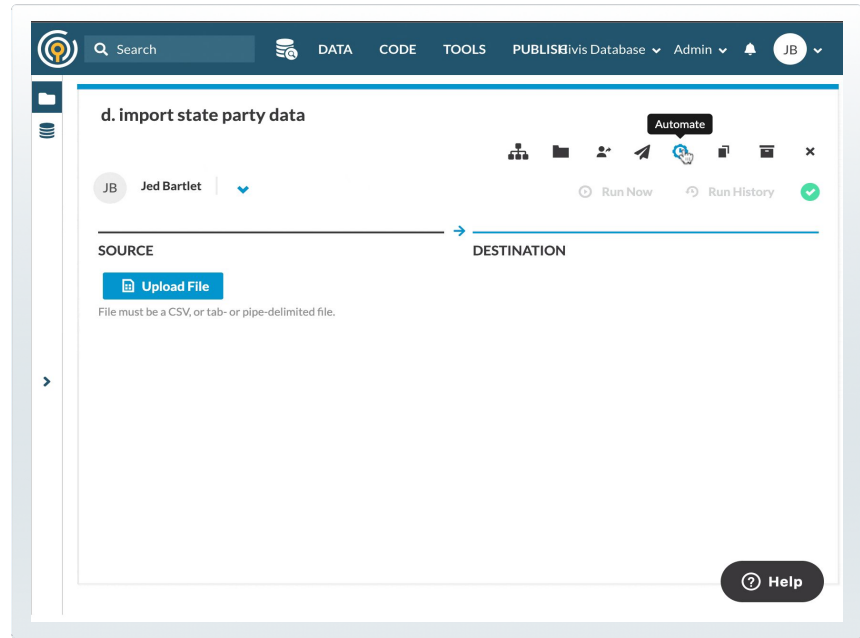


Scripts

Automation

- From the Standard Action Menu, select the “Automate” icon
- Toggle “Automate” to on
- Scroll through the automation options and choose the schedule

Note: the automation options may look slightly different depending on the type of Platform Object you are automating. For example, a SQL script allows you to automate by day/month/time, whereas a csv import allows you to automate by day/time



Scripts

Notifications

Job outputs in Notification Emails

It is possible to add files, markdown tables or json formatted data produced by your Job to the success notifications.

- For SQL Scripts you can add a link to download the output CSV to your email notification by using [link text]({{file_url}})
- We have example code in [Python](#) and [Javascript](#) that will convert a SQL query into a markdown table and topline number that can be added to your email

The screenshot displays a web interface for configuring a script and its notification settings. The main window is titled "Script #252810469" and shows the script's configuration, including the database name "Civis BigQuery" and the SQL query:

```
1 /* This scripts updates all reporting tables
2 and then runs a summary query to get topline
3 numbers for reporting to stakeholders */
```

Below the script configuration, a "Notify" modal window is open, showing the notification settings. The notification is configured to send an automated email upon success. The "On Success" toggle is set to "On". The "Email Recipients" field contains "civistestuser3@gmail.com". The "Email Subject" is "Today's Topline Numbers". The "Email Body" contains the following text:

Hey team,

The daily reporting tables update has completed and the summary query was run. Click [this link]({{file_url}}) to download the results as a csv.

The "Email Body" field also includes a note: "You can use markdown to add formatting." The "URLs" field is currently empty.



Version Control

Basics

Connecting Platform to GitHub

Setup

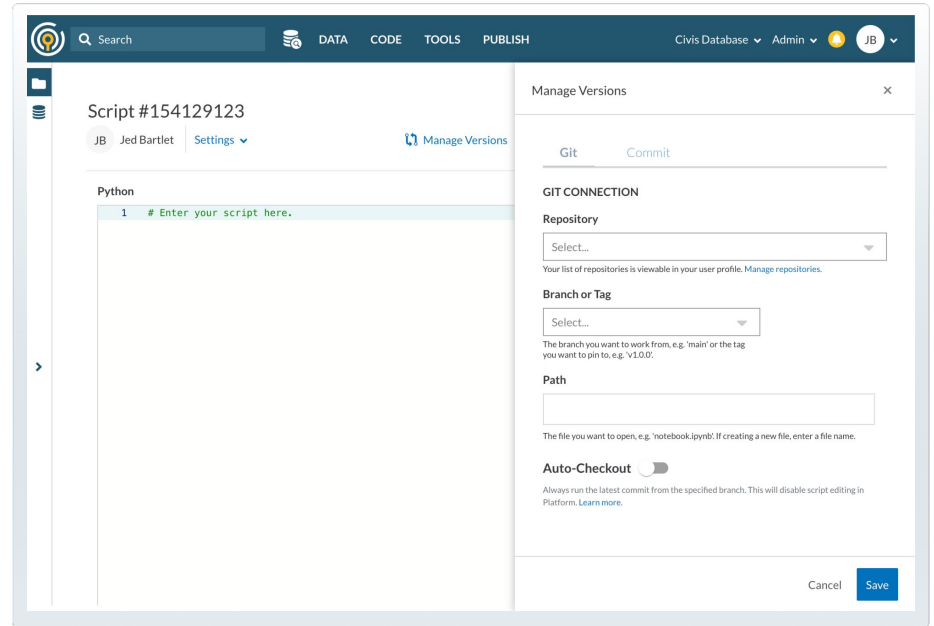
Commit, Push, and Checkout

Autocheckout

Version Control

Basics

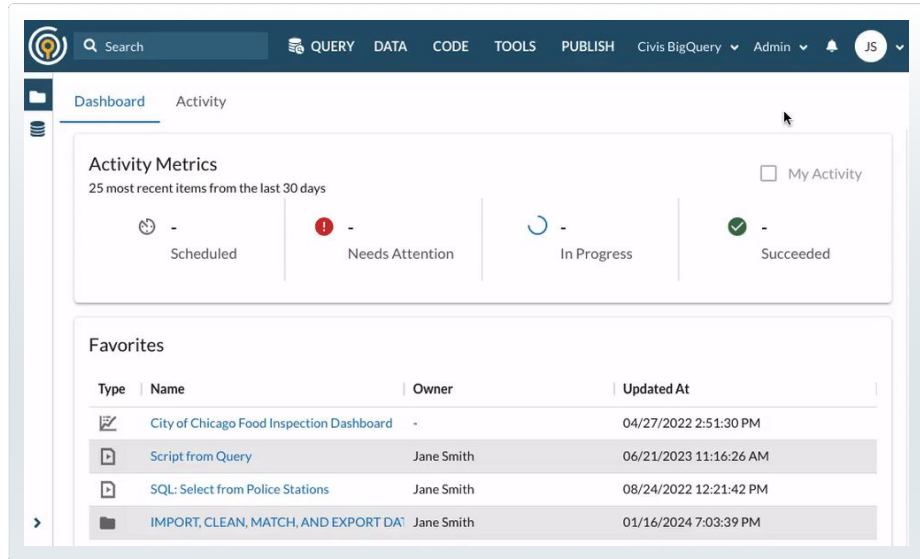
- Civis Platform supports version control for the following:
 - SQL, R, and Python Scripts
 - Workflows
 - HTML Reports
 - Notebooks
- Civis Platform supports version control via GitHub



Version Control

Connecting Platform to GitHub

- To connect to a GitHub:
 - Log into GitHub and have it open in another tab
 - Navigate to the Git Repos tab under My Profile
 - Click Add Repository
 - Click Connect to GitHub
 - If not automatically redirected to GitHub, click into your open GitHub Tab and then back to Platform
 - Search for and add a repository



The screenshot shows the 'Activity Metrics' dashboard. The top navigation bar includes a search bar, a 'QUERY' button, and menu items for 'DATA', 'CODE', 'TOOLS', 'PUBLISH', 'Civis BigQuery', 'Admin', and a user profile 'JS'. The main content area is titled 'Activity Metrics' and shows '25 most recent items from the last 30 days'. There are four status indicators: 'Scheduled' (clock icon), 'Needs Attention' (red exclamation mark icon), 'In Progress' (circular arrow icon), and 'Succeeded' (green checkmark icon). Below this is a 'Favorites' section with a table listing items.

| Type | Name | Owner | Updated At |
|------|---|------------|------------------------|
| | City of Chicago Food Inspection Dashboard | - | 04/27/2022 2:51:30 PM |
| | Script from Query | Jane Smith | 06/21/2023 11:16:26 AM |
| | SQL: Select from Police Stations | Jane Smith | 08/24/2022 12:21:42 PM |
| | IMPORT, CLEAN, MATCH, AND EXPORT DA | Jane Smith | 01/16/2024 7:03:39 PM |

Version Control

Setup

A. Manage Versions

to set up a Platform object for version control, click “Manage Versions”

B. Repository

a list of repositories your Git profile has access to

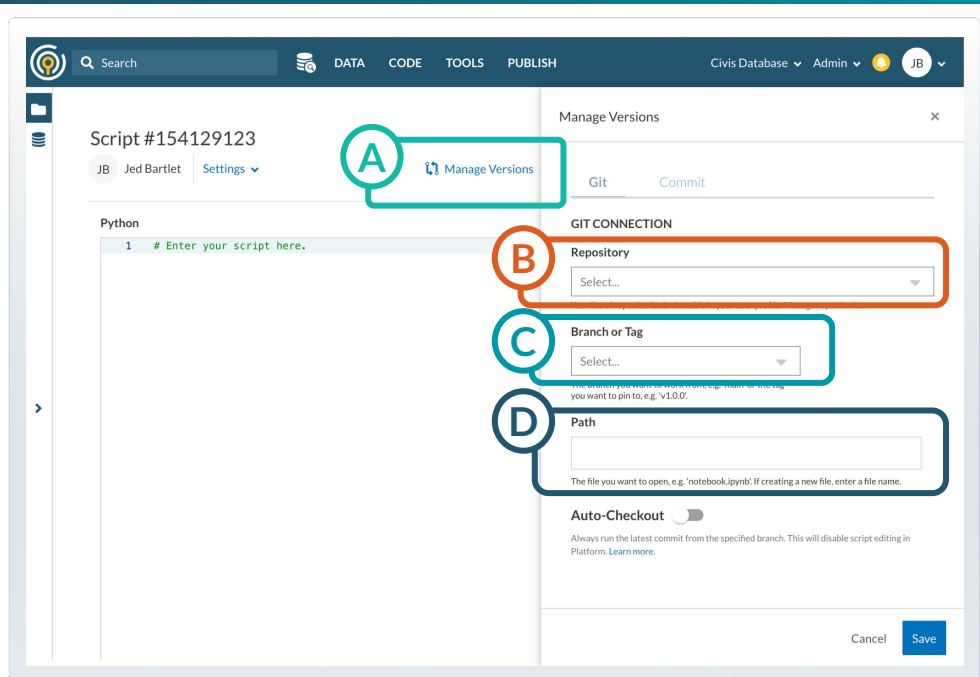
C. Branch

specific branch you would like to use. The branch must already exist in your repo

D. Path

If the file does not yet exist, Platform will create a new file for you in the repo.

Note: path must be a file, not a directory.



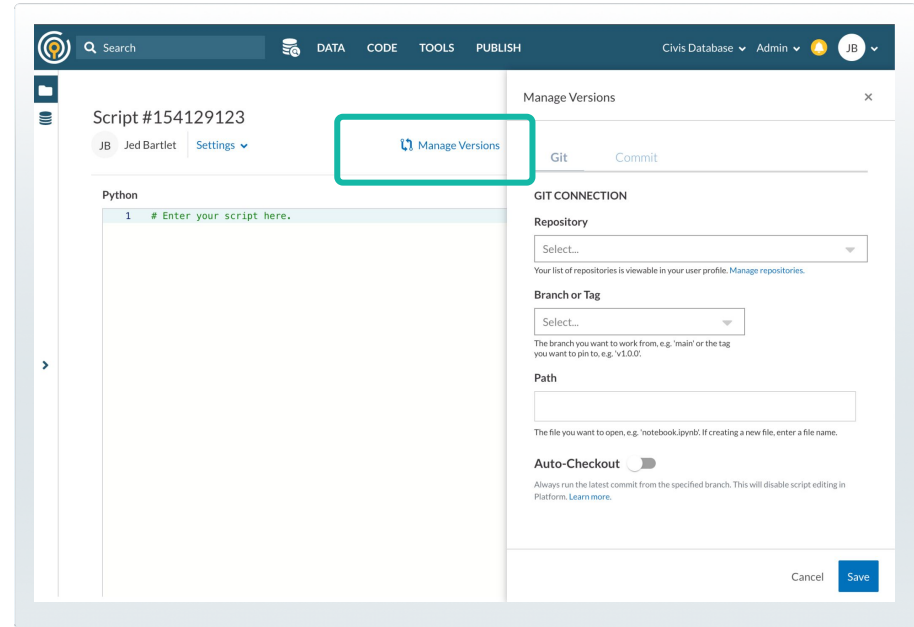
*Note: These steps apply to any existing or new SQL, R or Python script, workflow, HTML report, or stopped notebook.



Version Control

Commit, Push, and Checkout

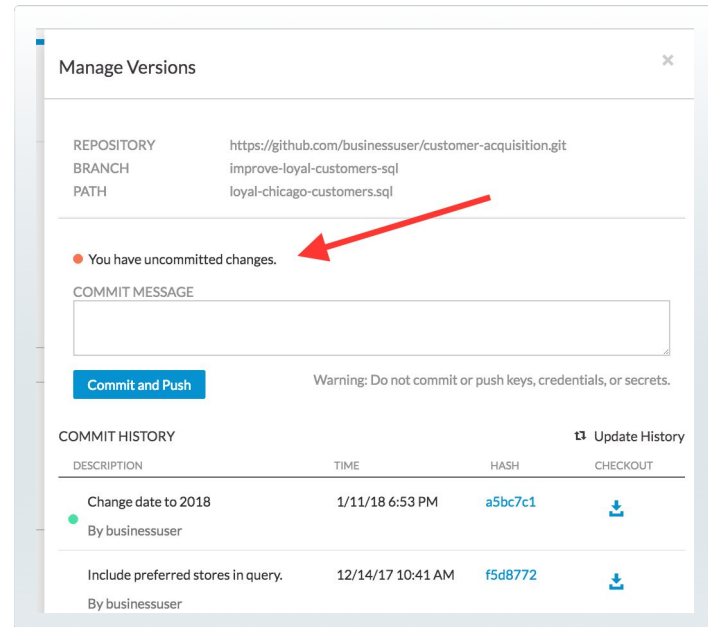
- After entering your repository's information, you can pull data from or push data to your repository through the Manage Versions pane
- While Platform automatically saves Platform objects, you will not see changes in Git until you manually commit and push these changes to Git repositories



Version Control

Commit, Push, and Checkout

- As you make iterative edits to your script in Platform, you can commit changes to the specified path
- Once you make edits to the script, an icon will appear that prompts you to “Commit Changes”
- Navigate to the “commit” tab of the Manage Versions pane, add a commit message and press “commit and push”
- Platform will not automatically commit and push these changes



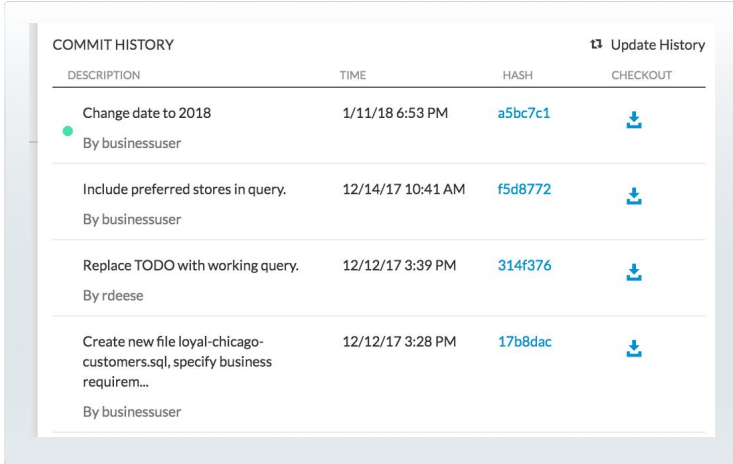
The screenshot displays the 'Manage Versions' interface. At the top, it shows repository information: REPOSITORY (https://github.com/businessuser/customer-acquisition.git), BRANCH (improve-loyal-customers-sql), and PATH (loyal-chicago-customers.sql). Below this, a red dot icon indicates 'You have uncommitted changes.', with a red arrow pointing to it. A text input field for the 'COMMIT MESSAGE' is present. A blue 'Commit and Push' button is visible, along with a warning: 'Warning: Do not commit or push keys, credentials, or secrets.' The bottom section, 'COMMIT HISTORY', includes an 'Update History' toggle and a table of previous commits.

| DESCRIPTION | TIME | HASH | CHECKOUT |
|---|-------------------|---------|----------|
| Change date to 2018 By businessuser | 1/11/18 6:53 PM | a5bc7c1 | |
| Include preferred stores in query. By businessuser | 12/14/17 10:41 AM | f5d8772 | |

Version Control

Commit, Push, and Checkout

- Platform will automatically pull a log of commits where the connected file was changed in the specific repository and branch selected
- The green dot in the Git commit log indicates if the current code matches that commit
- You can checkout any commit shown in the log
 - Platform will warn you if checking out old code will overwrite any uncommitted changes

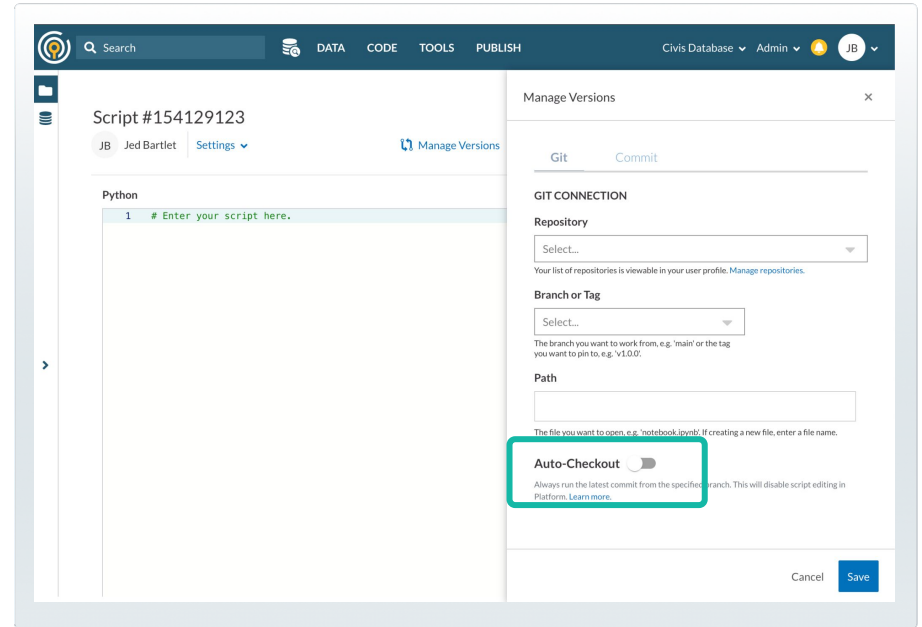


| COMMIT HISTORY | | | Update History |
|--|-------------------|---------|----------------|
| DESCRIPTION | TIME | HASH | CHECKOUT |
| ● Change date to 2018 By businessuser | 1/11/18 6:53 PM | a5bc7c1 | ↓ |
| Include preferred stores in query. By businessuser | 12/14/17 10:41 AM | f5d8772 | ↓ |
| Replace TODO with working query. By rdeese | 12/12/17 3:39 PM | 314f376 | ↓ |
| Create new file loyal-chicago-customers.sql, specify business requirem... By businessuser | 12/12/17 3:28 PM | 17b8dac | ↓ |

Version Control

Auto-checkout

- With Auto-Checkout Platform will always run the latest committed code from the specified branch of your repository.
- Editing script code in Platform will be disabled when Auto-Checkout is on to prevent conflicting changes
 - If you need to make changes to the script code, you can toggle Auto-Checkout off, make edits and commit the code.





Workflows

Overview

Creating New Workflows

Parameters in Workflows

Executing Workflows

Workflows

Overview



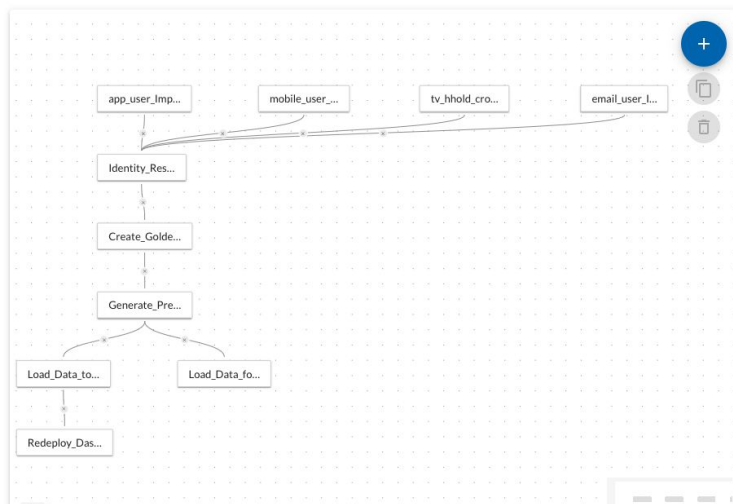
Platform Workflows give users the ability to chain together jobs to accomplish a specific task.

QSR Engagement Workflow

JS Jane Smith

Favorite Manage Versions History

Execute All



```
INFO      YAML      PARAMETERS
1  version: '2.0'
2  workflow:
3  input:
4    - start_date
5    - end_date
6  tasks:
7    app_user_import:
8      action: civis.run_job
9      input:
10       | job_id: 26009673
11     on-success: Identity_Resolution
12   email_user_import:
13     action: civis.run_job
14     input:
15       | job_id: 26009687
16     on-success: Identity_Resolution
17   tv_hhold_crosswalk_import:
18     action: civis.run_job
19     input:
20       | job_id: 26009694
21     on-success: Identity_Resolution
22   mobile_user_crosswalk_import:
23     action: civis.run_job
24     input:
25       | job_id: 26009696
26     on-success: Identity_Resolution
27   Create_Golden_Table:
28     action: civis.run_job
29     input:
30       | job_id: 26389512
31     on-success: Generate Predictions
```



Workflows

Overview

Graph and Info Tab

- Build Workflows using existing Jobs and Templates
- Create branching and basic task control flow
- Workflow Level Parameters for use with Templates

YAML

- Build Workflows using new or existing Jobs and Templates
- More advanced task control flow and conditional tasks
- Add retries to your Tasks
- Workflow level parameters can be used in new Jobs and with Templates
- Execute existing Workflow as a Workflow task
- And more

Workflows

Overview

- The Graph is a visual representation of your workflow
- Changes made in the Graph UI or in the YAML will automatically update the other to reflect your changes.
- The workflow runs from top to bottom, traveling along the links between tasks
- Platform Workflows are built on the open source Mistral workflow engine
- You can create/edit workflows using the graph on the left and the INFO tab or via the Mistral YAML-based workflow language in the expandable YAML tab

Graph & Yaml Example

JS Jane Smith

Favorite Manage Versions History

```
20 inputs
21 | job_id: @
22 on-success: ETL
23 on-error: pipeline_fail
24 ETL:
25 action: civis.run_job
26 input:
27 | job_id: @
28 join: all
29 on-success: data_QC
30 on-error: pipeline_fail
31 data_QC:
32 action: civis.run_job
33 input:
34 | job_id: @
35 on-error:
36 - data_fail
37 - pipeline_fail
38 on-success: refresh_report
39 data_fail:
40 action: civis.run_job
41 input:
42 | job_id: @
43 pipeline_fail:
44 action: civis.run_job
45 input:
46 | job_id: @
47 join: all
48 refresh_report:
49 action: civis.run_job
50 input:
51 | job_id: @
52 on-error: pipeline_fail
53
```

Workflows

Overview

Workflow Control Flow

- On-success
 - Next task(s) will run after this task completes successfully
- On-complete
 - Next task(s) will run after this task completes regardless of state
- On-error
 - Next task(s) will run after this task completes unsuccessfully
- Join
 - Determines if multiple parent tasks must complete before child task runs
 - Workflows created via UI defaults to a Join: All on all child tasks
 - Without Join: All the child task may run more than once



Workflows

Graph View Terms to Know

A. Workflow

a collection of related jobs intended to be run with specific dependencies on each other.

B. Task

a unit of work inside of a workflow.

C. Action

the job that will be run at a particular moment during a workflow.

D. Execution

a single run of a workflow.

Data Cleaning and Update Report Workflow

JB Jed Bartlet

Favorite | M | D | History | Execute All

INFO | YAML | PARAMETERS

Task Name: GOOGLE_ANALYTICS_IMPORT

Task Details

Action: civis.run_job

Job Name: Google Analytics Import: Pages (Fundraising) (ID:182142...)

Comments

Parent Task Connections

On Success: [Dropdown]

On Error: [Dropdown]

On Complete: [Dropdown]

Child Task Connections

On Success: DATA_CLEANING

On Error: [Dropdown]

On Complete: [Dropdown]

```
graph TD; SALESFORCE_I[SALESFORCE_I...] --> DATA_CLEANIN[DATA_CLEANIN...]; GOOGLE_ANALY[GOOGLE_ANALY...] --> DATA_CLEANIN; EVERYACTION[EVERYACTION...]; DATA_CLEANIN --> ADDRESS_CORR[ADDRESS_CORR...]; ADDRESS_CORR --> IDR[IDR]; IDR --> GOOGLE_SHEET[GOOGLE_SHEET...]; IDR --> REFRESH_TABL[REFRESH_TABL...]; IDR --> SALESFORCE_E[SALESFORCE_E...]; IDR --> NGPVAN_EXPOR[NGPVAN_EXPOR...];
```

Workflows

Graph View Terms to Know

E. Parent Task(s)

The task(s) that the currently selected task will run after when the run condition is met

F. Child Task(s)

The task(s) that will run after the currently selected task when the run condition is met

Data Cleaning and Update Report Workflow

JB Jed Bartlet

Favorite Manage Versions History Execute All

INFO YAML PARAMETERS

Task Name
GOOGLE_ANALYTICS_IMPORT

Task Details

Action
civis.run_job

Job Name
Google Analytics Import: Pages (Fundraising) (ID:182142...)

Comments

Parent Task Connections

On Success

On Error

On Complete

Child Task Connections

On Success
DATA_CLEANING

On Error

On Complete

Workflows

YAML Terms to Know

A. Task Name

The name of the task, for example, “Python”

B. Task Action

API Call Endpoint

C. Task Input

Information that are passed into the API endpoint called

D. Child Task(s)

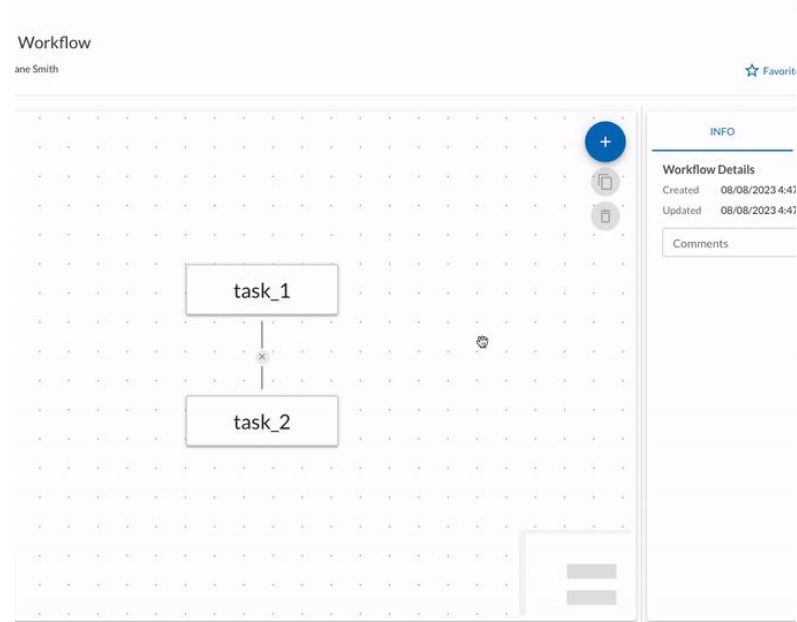
Names of the tasks to run once current task completed, if the run condition is met

```
1  version: '2.0'
2  workflow:
3  tasks:
4  python:
5    action: civis.scripts.python3
6    input:
7      name: python is the best
8      source: print("Hello from Python!")
9    on-success:
10     - r
11  r:
12    action: civis.scripts.r
13    input:
14      name: R is the best
15      source: print("Hello from R!")
16    on-success:
17     - container
18  container:
```

Workflows

Creating a New Workflow: INFO Tab

- Click the + sign in the upper right, if you have a task selected already the new task will become a child
- Find existing Jobs using the Job Name Field
- Delete a task by clicking the trash can icon
- Click and drag one of the blue dots from one task to another task to chain the two together
- To unchain two jobs, click the X on the line connecting the task



Workflows

Creating a New Workflow: YAML Tab

- The YAML tab shows the workflow definition in YAML form
- The workflow contains a list of tasks, each with a unique name
- Each task specifies an action (e.g. running a job), an input list (e.g. the ID of the job to run), and an optional list of other tasks that should be run immediately following the success, completion, or failure of this task (on-success, on-complete, on-error)

```
INFO                                YAML
1  version: '2.0'
2  workflow:
3    tasks:
4      python:
5        action: civis.scripts.python3
6        input:
7          name: python is the best
8          source: print("Hello from Python!")
9        on-success:
10         - r
11
12      r:
13        action: civis.scripts.r
14        input:
15          name: R is the best
16          source: print("Hello from R!")
17        on-success:
18         - container
19
20  container:
21
```

Workflows

Creating New Workflow: YAML Tab

- Users can also refer to existing Platform jobs and workflows in the Mistral workflow language
 - Existing jobs use the `civis.run_job` task action
 - This action takes a single input: `job_id`.
- Existing workflows can be executed within another workflow
 - Use the `civis.workflow.execute` task action to execute
 - This action takes in a single input: `workflow_id`.

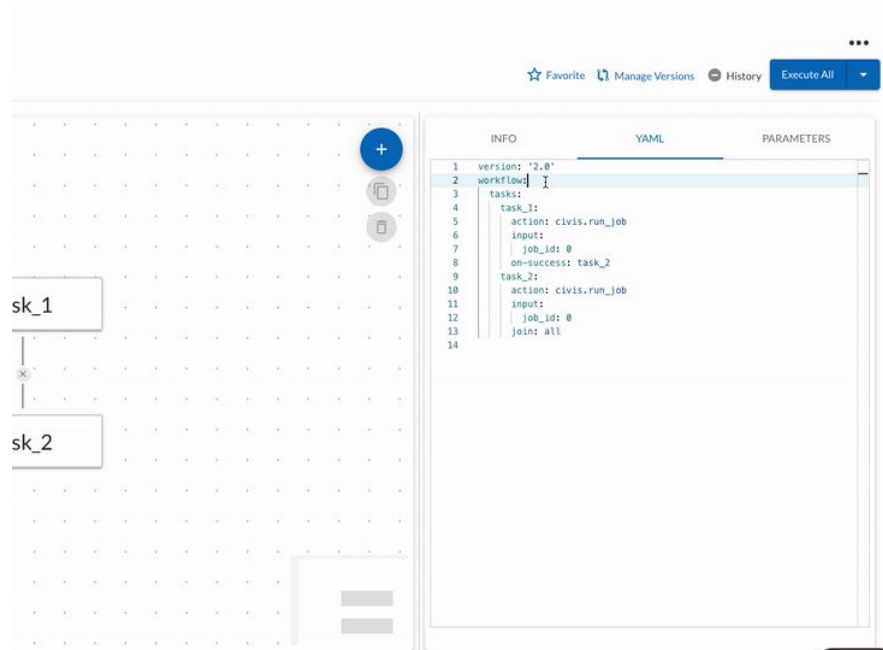
```
tasks:  
  task_a:  
    action: civis.run_job  
    input:  
      job_id: 34
```

```
tasks:  
  execute_subworkflow:  
    action: civis.workflows.execute  
    input:  
      workflow_id: 56
```

Workflows

Parameters in Workflows

- Use the Parameters tab, to create new parameters or see parameters declared in YAML
- In the input fields that appear, enter values for your parameters
- You can supply default value in the Parameter tab or YAML, it will appear pre-populated in the input field
- Each parameter input will be interpreted as a string
- Parameters are accessed via YAQL like this `<% $.parameter_name %>`



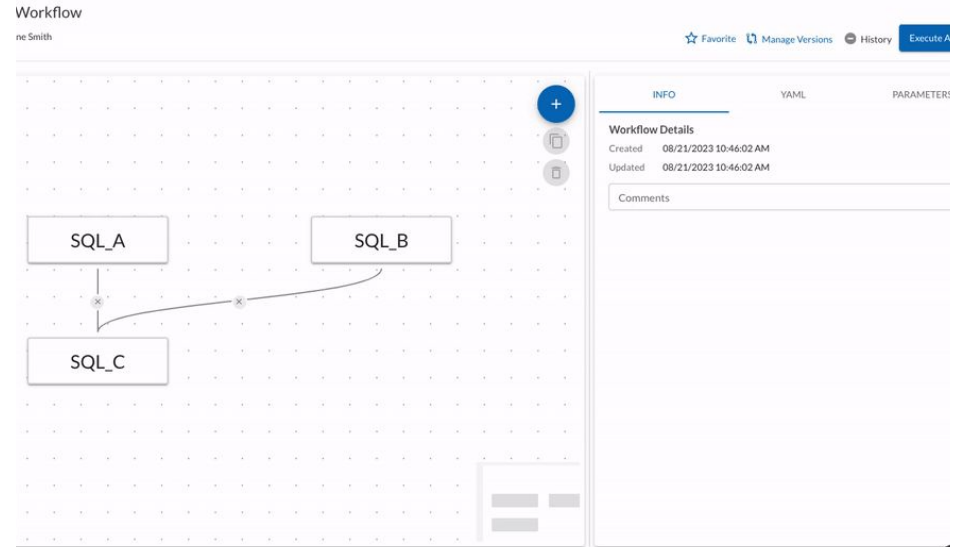
The screenshot displays the Civis Workflows interface. On the left, there is a grid-based editor with two input fields labeled 'sk_1' and 'sk_2'. On the right, the 'PARAMETERS' tab is active, showing a YAML configuration for a workflow. The YAML code is as follows:

```
1 version: '2.0'
2 workflow: |
3   tasks:
4     task_1:
5       action: civis.run_job
6       input:
7         job_id: @
8       on-success: task_2
9     task_2:
10      action: civis.run_job
11      input:
12        job_id: @
13      join: all
14
```

Workflows

Executing Workflows

- To run a workflow, click “Execute (All Tasks)”
 - If Execute is grayed out, there is a problem with the workflow definition
- Once running, the workflow graph will display the execution and highlight when tasks have run successfully or failed
- You will be able to see the Job logs of a selected task in the INFO tab



Workflows

Executing Workflows: Partial Executions

- To execute only some tasks click a task and then hover over other tasks so that the check box to their right appears and select that box to include that task
- When you have the tasks you wish to execute selected click the “Execute Selected Tasks” button
- When the execution loads tasks not included will be greyed out

The screenshot displays a workflow editor interface. At the top, the title "workflow" is visible, along with a user name "Smith" and a menu with options: "Favorite", "Manage Versions", "History", and "Execute All". The main workspace is a grid where three tasks are arranged: "SQL_A" and "SQL_B" are at the top, and "SQL_C" is below "SQL_A". A curved arrow connects "SQL_A" to "SQL_B", and a straight arrow connects "SQL_A" to "SQL_C". Small 'x' icons are present on the arrows. On the right side of the grid, there are three circular icons: a blue plus sign, a grey square, and a grey circle. A "Help" button is located in the bottom right corner of the grid area. To the right of the grid is a "Workflow Details" panel with tabs for "INFO", "YAML", and "PARAMETERS". The "INFO" tab is active, showing "Created 08/21/2023 10:46:02 AM" and "Updated 01/09/2024 10:16:47 AM". Below this is a "Comments" text area.

Workflows

Executing Workflows

- If your Workflow fails you can use the re-try failed tasks button. Your workflow will retry any failed tasks and resume execution
- Use “History” to see the status of a Workflow’s previous executions
- Clicking on any of these executions displays a more detailed view in the main window, where you can see the graph of tasks and the Workflow definition that the execution was run with

The screenshot displays a workflow execution interface. At the top left, the user is identified as 'Jane Smith'. On the top right, there is a 'History' button with a red exclamation mark icon and a blue 'Retry Failed Tasks' button. The main area shows a workflow graph with three tasks: 'Task_A' (green checkmark), 'Task_B' (green checkmark), and 'Task_C' (red exclamation mark). A curved arrow points from the failed 'Task_C' back to 'Task_B'. To the right of the graph, there is a sidebar with 'INFO' and 'YAML' tabs, and a 'Workflow Details' section containing the following information:

- Created: 01/30/2024 5:37:07 PM
- Updated: 01/30/2024 5:38:33 PM
- Comments: -

Below the graph, there is a 'Status' section showing 'Failed' with a red exclamation mark icon, and a 'Duration' of '0 hr 00 min 13 sec'. An 'Info' dropdown menu is also visible.



Thank you!

Reach out to Support@CivisAnalytics.com with any questions!

Our next webinar will be with our Product Team on new features and updates to Platform. Watch for the sign up form!