CHAPTER

SIXTEEN

PROGRESS DISPLAY

Rich can display continuously updated information regarding the progress of long running tasks / file copies etc. The information displayed is configurable, the default will display a description of the 'task', a progress bar, percentage complete, and estimated time remaining.

Rich progress display supports multiple tasks, each with a bar and progress information. You can use this to track concurrent tasks where the work is happening in threads or processes.

To see how the progress display looks, try this from the command line:

```
python -m rich.progress
```

Note

Progress works with Jupyter notebooks, with the caveat that auto-refresh is disabled. You will need to explicitly call *refresh()* or set refresh=True when calling *update()*. Or use the *track()* function which does a refresh automatically on each loop.

16.1 Basic Usage

For basic usage call the *track()* function, which accepts a sequence (such as a list or range object) and an optional description of the job you are working on. The track function will yield values from the sequence and update the progress information on each iteration. Here's an example:

```python
import time
from rich.progress import track

for i in track(range(20), description="Processing..."):
    time.sleep(1)  # Simulate work being done
```

16.2 Advanced usage

If you require multiple tasks in the display, or wish to configure the columns in the progress display, you can work directly with the *Progress* class. Once you have constructed a Progress object, add task(s) with (*add_task()*) and update progress with *update()*.

The Progress class is designed to be used as a *context manager* which will start and stop the progress display automatically.

Here's a simple example: