# Coders and ODP Staff in Additional Training Instances

1. Each of the nine coders (the new cohort plus Lamyaa, Daniela, and Shahina) are each given their own fresh Instance of the training environment. This would be the same environments that we used two weeks ago to test the new cohort.

2.      IQ pulls 300 random un-coded abstracts from Production and loads them into the Instances such that all Instances show the abstracts in the same order.

3.      For each abstract:

a.      Each coder conducts Individual Coding

b.      The robots input E7F6

c.       A group of three coders then convenes and develops a consensus for that abstract

d.      Each coder then individually inputs that shared consensus into the Consensus Coding of their individual Instance

e.      They then move on to the next abstract

f.        This means that coders get the experience of negotiating a consensus inside of a group, but without the Watch Consensus feature

g.       After Coder inputs Consensus, the Coder Supervisor uploads notes for that abstract, moving the abstract from 1B to 1N.

4.      As one of the coders completes an abstract, the ODP team logs in to that Instance and codes that same abstract through to Comparison/Reconciliation. This means that we will effectively have an answer key for this set of abstracts.

1. The ODP staff will choose a single of the Instances to code; the goal being to get abstracts to Status 2C.

As ODP completes the coding of abstracts on a single Instance (thereby generating an answer key), we need some method to populate those answers across the other 8 active Instances. That population process is described below in the section **“Process to Populate ODP selections across all instances”.**

Finally, we generate Kappas across all of these instances. This gives us final scores for all of the coders for this (ad hoc) test set #2. That method is described below in the section **“Process to Generate Kappas across all instances”.**

**Note:** We agreed that in the target Instances, we would treat the ODP team as actual coders rather than robots. This requires that the same group of three ODP coders code all abstracts in the source Instance.

**Note:** In the current system, the function for creating the team is that the largest UserID is CoderA, the second largest is CoderB, and the smallest is CoderC. In the Virtual environments, some real people have UserIDs that are smaller than the Robots’ UserID; so, in some cases, the real person will be CoderC rather than CoderA. We will add User Name to the report so that ODP knows which Coder is CoderA or CoderC for each Instance.

# Fill in ODP selections across all instances

Once ODP has coded behind the abstracts in a single instance, they need some method to populate those completed abstracts from that single Instance across to all of the other Instances. They will do this via a web interface similar to that at <http://odptaxtrainingadmin.iqsolutions.com>.

1. ODP user logs into the system
2. Using a single selection dropdown box, User chooses which Instance they want to populate answers to. This is the **target instance.** The **source instance** should be configured into the system, since that will not change.
3. User presses a button labeled “Populate ODP Selections”.
4. The system populates the answers from the source instance **to the Main Database and then from the Main Database** to the target instance. The system then issues a success message “Selections populated to Instance [$N]”
   1. This “success” screen should also include a list of the abstracts populated – APPL ID, PI Name, and Title – sorted by APPL\_ID ascending.
   2. The population process should only populate an abstract when it has reached status 1N in the **target system** and status 2C in the **source system.**
   3. **The purpose of replicating to the Main DB before pushing to the Instance is to allow the Main DB to serve as a long-term repository of all ODP selections for this set of abstracts. In the future, when we reuse and recycle Instances, we can populate that Instances ODP selections from Main DB.**
5. The ODP Supervisor must complete this process for each target Instance individually, rather than as a single process that populates all target Instances in one button press.

# Process to Calculate Kappas across all instances

Once ODP has coded behind the abstracts in a single instance, and populated those answers across to the target Instances, they need some method to generate Kappas 1-12 for all 9 Instances. They will do this via a web interface similar to that at <http://odptaxtrainingadmin.iqsolutions.com>.

1. ODP user logs into the system
2. Using a single selection dropdown box, User chooses which Instance they want to populate answers to. This is the **target instance.** The **source instance** should be configured into the system, since that will not change.
3. User presses a button labeled “Push Trainee Data”. The system pushes data from that Instance to Main for Kappa generation of all 12 kappas, then issues a success message, including instructions to wait for the next quarter-hour to pass before taking the next step.
4. User then presses a button labeled “Pull Trainee Kappa”. The system pulls kappa results from Main back to the Instance, then issues a success message.
5. To view Kappas for that target Instance, the ODP user must log into that Instance as ODP Supervisor and view the Kappas.