**update\_conf.sh**

The following script reads each line of the sig.conf file, modifies the line that matches the specified view, scale, and component name, and then writes the modified lines back to the file.

#!/bin/bash

# Function to validate user input

validate\_input() {

local input=$1

local options=$2

for option in ${options//\// }; do

if [[ "$input" == "$option" ]]; then

return 0

fi

done

return 1

}

# Ask for user inputs

read -p "Enter Component Name [INGESTOR/JOINER/WRANGLER/VALIDATOR]: " component

validate\_input "$component" "INGESTOR/JOINER/WRANGLER/VALIDATOR" || { echo "Invalid input."; exit 1; }

read -p "Enter Scale [MID/HIGH/LOW]: " scale

validate\_input "$scale" "MID/HIGH/LOW" || { echo "Invalid input."; exit 1; }

read -p "Enter View [Auction/Bid]: " view

validate\_input "$view" "Auction/Bid" || { echo "Invalid input."; exit 1; }

read -p "Enter Count [single digit number]: " count

if ! [[ "$count" =~ ^[0-9]$ ]]; then

echo "Invalid input."

exit 1

fi

# Temporarily store the modified lines

modified\_lines=()

while IFS= read -r line; do

if [[ "$line" == "$view ; $scale ; $component ; ETL ; vdopia-etl="\* ]]; then

modified\_line=$(echo "$line" | sed "s/vdopia-etl=[0-9]\+/vdopia-etl=$count/")

modified\_lines+=("$modified\_line")

else

modified\_lines+=("$line")

fi

done < "sig.conf"

# Write the modified lines back to the file

printf "%s\n" "${modified\_lines[@]}" > "sig.conf"

echo "File updated successfully."