04\_CommandReference

# Production Commands

## add transform

Adds a transform to the current working sequence. You may specify either the name or numeric ID of a transform. This is equivalent to selecting a transform from the numbered console menu.

Example:

add transform SlidingMaskOverlayTx

## clear sequence

Clears the currently assembled transform sequence. Use this if you want to start fresh when assembling a new pipeline.

Example:

clear sequence

## exit

Exits the workbench.

Example:

exit

## find best sequence

Tests all permutations of the current sequence to determine the best ordering based on cryptographic scoring. This command is useful when order matters but the transform list is known.

Example:

find best sequence

## get

Retrieves the value of a current workbench setting. Useful for confirming what is active during a session.

Example:

get Rounds

## help

Displays the command help screen.

Example:

help

## list

Lists all global configuration values currently active, including InputType, Rounds, Scoring Mode, and more.

Example:

list

## optimize sequence

Tunes the transform rounds (TR) and global rounds (GR) for the current sequence to find its optimal cryptographic performance. This is the most reliable way to fine-tune a known-good sequence.

Example:

optimize sequence

## optimize sequence gr

Tunes only the global rounds (GR) while leaving per-transform rounds unchanged. Useful if you want to evaluate GR sensitivity independently.

Example:

optimize sequence gr -max 9

## run comparative analysis

Executes a side-by-side comparison of the current Mango sequence vs AES using the selected InputType. Useful for proving superiority or benchmarking differences.

Example:

run comparative analysis

## run munge

Launches an automated discovery session to search for the best sequences for a given InputType. This is the foundation of how Mango discovers ‘god-sequences’.

Example:

run munge

## run regression tests

Executes the built-in regression suite to verify that transforms, scoring, and reversibility all behave as expected. Run before release or after large changes.

Example:

run regression tests

## run sequence

Encrypts, decrypts, and scores the current transform sequence using the selected InputType. Results are shown with detailed metric breakdown.

Example:

run sequence

## set

Updates a global configuration setting, such as `Rounds`, `InputType`, or `Mode`. Used to prepare the environment before running sequences.

Example:

set Rounds 10

# ⚙️ Debug & Experimental Commands

These commands are available only in Debug builds or developer environments. They provide deep access to internal optimizations, benchmarking, and SQL export tools.

## batch optimize sequences

Performs transform round (TR) and global round (GR) optimization in batch mode across multiple input sequences.

## batch optimize + reorder sequences

Optimizes and reorders sequences in batch mode, testing different transform orders alongside TR/GR tuning.

## log to screen

Displays the top contender sequences and their metrics in the console window after a Munge.

## log to file

Writes all contenders from the last Munge run to a text file for archival or analysis.

## log to sql

Exports all contenders from the last Munge run to a SQLite `.db` file.

## convert file to sql

Takes a previously generated contender log and converts it into a `.db` file.

## query

Retrieves values for internal metrics or system scoring weights.

## run query console

Launches the SQL console for exploring and debugging output databases.

## run analyzer

Runs internal scoring/metric analysis routines.

## run auto weight tuner

Experimental tool for tuning metric weights automatically.

## run benchmark transforms

Benchmarks each transform's runtime for internal profiling.

## run profile transforms

Collects statistical data for each transform’s behavior under test input.

## run MangoCipher

Runs the Mango cryptographic engine directly outside the Workbench pipeline.

## run munge e

Executes Munge in Exploratory Mode.

## run munge k

Executes Munge with a focus on key sensitivity and variation.

## run smart munge

Experimental Munge system that adapts based on early results.

## run visualization

Renders a visual heatmap or histogram of sequence output.

# 🧩 Special Console Symbols

## $ (Paste Sequence)

Use `$` to paste an entire transform sequence in one step.

Example:

$MaskedCascadeSubFwdFbTx -> SubBytesInvTx -> ApplyMaskBasedMixingTx

## ! (Bang Command)

Typing `!` alone will clear the current sequence and auto-load the god-sequence for the selected InputType.

You may also append `!` to commands to use the god-sequence implicitly:

run sequence !

This example clears the current sequence, loads the best god-sequence for the active InputType, and runs it immediately.