Run.Net, revised [Mazu.Net]

Run.Net is a semantic command-line system for simulating ‘Netrunning’ in your Cyberpunk games. Actions in Run.Net are created ‘semantically’...if you wish to conceal a User’s presence on a system, you use the Command “Conceal” on the Target “User”, and as a player, you say “Conceal User” in your declaration. This avoids unnecessary crunch, keeps Netrunning actions as free-form in their meaning as the other skills in the Interlock system, and keeps the system ‘agnostic’ of the hardware background you’re running with.
If you wish to simulate the sense of a particular kind of hardware, we advise you research it for its peculiarities, and then use those concepts in play. A Babbage engine will be immune to EMP - but a Quantum Computer might do something really strange without crashing!

Interface Skill

The Interface Skill represents the character’s ability to manipulate computers. Interface can be used in the normal system of Stat + Skill + D10 for simple tasks, in the same way you could use Tech + Drive + 1D10 to drive home.
For more complex actions you need to use the Command & Target lists.

Command & Target

Commands are functions a User can perform, and Targets are what the commands are performed on. In this text, we will use the format [Command//Target].

> For example:
> [Conceal//User]
> [Edit//File]
> [Infiltrate//Modem]

Another format can be found in the text. If a user does an action that is missing either the command or target then the action will be in single quotes (‘example’).

> For example:
> The user ‘Conceals’
> The runner ‘Scans”
> The Intruder ‘Infiltrates’

Since the Target for these actions can depend on the situation the single quote is used. This means that a Command//Target Action is being used but either the command or the target can vary depending on the situation.

The Command List:

1. **Detect/Conceal** - To determine presence of Target or overcome concealment / to hinder discovery or disguise to avoid detection by another party.
2. **Locate** - To find physically or Digitally within a given set of parameters.
3. **Infiltrate/Login** - To gain unwarranted access to or bypass security of Target.
4. **Control** - To execute normal operation of Target.
5. **Secure** - To prevent infiltration, modification, or control of Target by outside force.
6. **Cypher** - To render a File or Signal indecipherable to another party / To decipher an encrypted File or Signal.
7. **Scan** - To assess the condition of a Target. To maintain passive observation of a Target.
8. **Query** - Request specific information that can be gained from Target. Serial number, version, manufacturer, size, hardware linkage, range, mode of operation, etc.
9. **Edit** - To alter a Program, File, or Database.
10. **Run Program** - Execute software from a controlled system.

To do an action using the Command//Target System
Stat + Command + Target + Mods+ D10

The Target List:

1. **File/Database** - Set of data contained as a unit for use by program, system, or user.
2. **Cyber** - Hardware that utilizes bio-mechanical interface technology.
3. **Comm.** - Technology designed to enable the communication of 2 or more parties (e.g. phones).
4. **Sensor** - Hardware designed to gather sensory data (Camera, Microphone, Touch/Heat/Motion sensor, etc).
5. **Remote** - Mobile system operating under control of user or system from a distance.
6. **Modem** - Hardware designed to interface a user or system with the Net.
7. **Vehicle** - System designed to physically transport user or materials.
8. **Weapon** - System designed to cause physical damage.
9. **System** - CPU in control of a given set of parameters.
10. **User/Runner** - Human (or AI) operator connected to a system or hardware.

To do any task online, you should combine a Command and a Target in your declaration to the GM.

"I Conceal User, then I'll roll to Infiltrate Modem."

"I want to add a myself to that delivery database, so… Edit Database, and add a new line to deliver a Ninja 750 to my pickup address."

"Locate Comm on frequencies for mobile devices. I want to find that guy before he

gets out of the club."

Interface x 5

Multiply your Interface Skill by 5. These points represent your familiarity with the tasks and systems of a computer. System Familiarity Points = Interface x 5. Distribute your Total Familiarity Points to the Command list and the total again to the Target list. This represents your character's skill at performing those tasks and their familiarity with those targets.
Command//Target Skills are capped at 5, it is not possible to go higher. If your Command and/or Target skill is 0 you can still attempt the task - you're guessing your way through the interface, clicking on menus that look useful, or even using the help files.

Chas Kingston (Net Handle: BlueSteel) has an Interface of 3. This means his System Familiarity Points total is 15. That means 15 points to the Command list, and 15 points to the Target List. He chooses to the commands: Detect/Conceal 3, Infiltrate/Login 3, Control 2, Cypher 2, Scan 2, Query 2, and Edit 1. A total of 15 points. His target list is Comm 3, Sensor 3, Remote, 3, Modem 4, and System 2. A total of 15 points.

Examples of Command Skills:
Competent office staff would have a high 'Edit' skill. A competent Hacker will have a high 'Infiltrate.' A spy might use 'Locate' a lot to track targets. A double agent will want 'Cipher' to protect their files.

Examples of Target Skills:
Why the cap of 5? So that we don't 'break' Interlock by pushing past the normal range of 0 to 10. When you combine your Command & Target skill, you are recomposing the Interface skill that was broken into detailed parts, and that is what fits in the normal Stat + Skill + D10 function.

A MedTech installing bionics will want a high 'Cyber' to personalize settings for their patient.
A Miner using robotic diggers will use 'Remote' to control his machines. A

Signals Officer will use ‘Comm’ to keep information flowing on the Net-Centric battlefield.
A Netrunner will want to be familiar with ‘User’ to kick hostile Runners out of the battle.

What happens when I increase my Interface?

Ding! Say you’ve gone from Interface 3 to 4. You now have 5 Familiarity points to spend on the Command list, and on the Target List. Yes, 5 to each. Every time the characters Interface skill increases they get 5 points to spend on each list.

Actions - Basic

We mentioned semantic command line Netrunning before - what this means is that your declaration of an action has already encoded the Command & Target component of your roll. Succeed and you get the result you want! Fail, and your opponent becomes aware something is going on. Other Users might notice flashing modem lights, grinding drives, or lurches in system performance. They don’t know what, but they know something is going on.

Example Actions:
“I infiltrate their modem.” = [Infiltrate//Modem] vs Firewall
“I delete that rap sheet.” = [Edit/Database]
“I aim the sentry gun away from me.” = [Control//Weapon]
“I scramble their comms.” = [Cypher//Comm]
“I give the UAV a new waypoint.” = [Control//Remote(Waypoints)]
“I hide my presence.” = [Conceal//User(self)]
“I hide the database.” = [Conceal//Database]

Rolling To Hit:
Run.Net’s Command//Target system uses a modified version of the basic Stat + Skill +D10. Since the Interface Skill is broken down into Command and Target Lists the same applies to rolls. Use **Int +Command + Target + Mods + 1D10 vs DV** to determine success.

DV (Difficulty Value) can be an opposed roll, or a fixed difficulty. Anything from a Software Firewall to an opponent’s persistent action.

If you fail an action then everyone on that system gets to make an immediate ‘Detect’ roll. If you are not concealing your actions the DV to detect is 5. Else the DV is equal to or greater than your previous ‘Conceal’ check. If the other users fail their ‘Detect’ they are aware that “something” has happened but are madly digging through screens of information, frustrated by doubt, but are unsuccessful in finding you or what happened.

On a success, they become aware of what just occurred, and can then begin to counteract it at the start of their turn.
(This replaces ‘Awareness’ rolls, only in the Net.)

Multiple Actions

So you’ve infiltrated, and you immediately want to Query that Database? Just roll another basic action, at a cumulative -2 penalty. This replaces the -3 for real-world multiple actions, because the Net is designed with speed and ease in mind. Want to combine actions? Say you need to get access to the Security Camera right this instant!
An example might be :
[Infiltrate//System] > to get into the Security control board. [Control//Remote] -2 > to take over the Security Camera. The camera shows a Guard is coming down the hall, so you shoot him. Now you’re taking a Real-World action, and it’s on top of net actions, you apply the Real-World -3 modifier for a total of -5.
Stack the penalty in the order they happen - it might make more sense to take a Real World action first while it is still easy to overcome the -3, and stack on Net actions at -2 afterwards.

You may attempt no more Actions per turn than your Interface skill.

Actions - Persistent

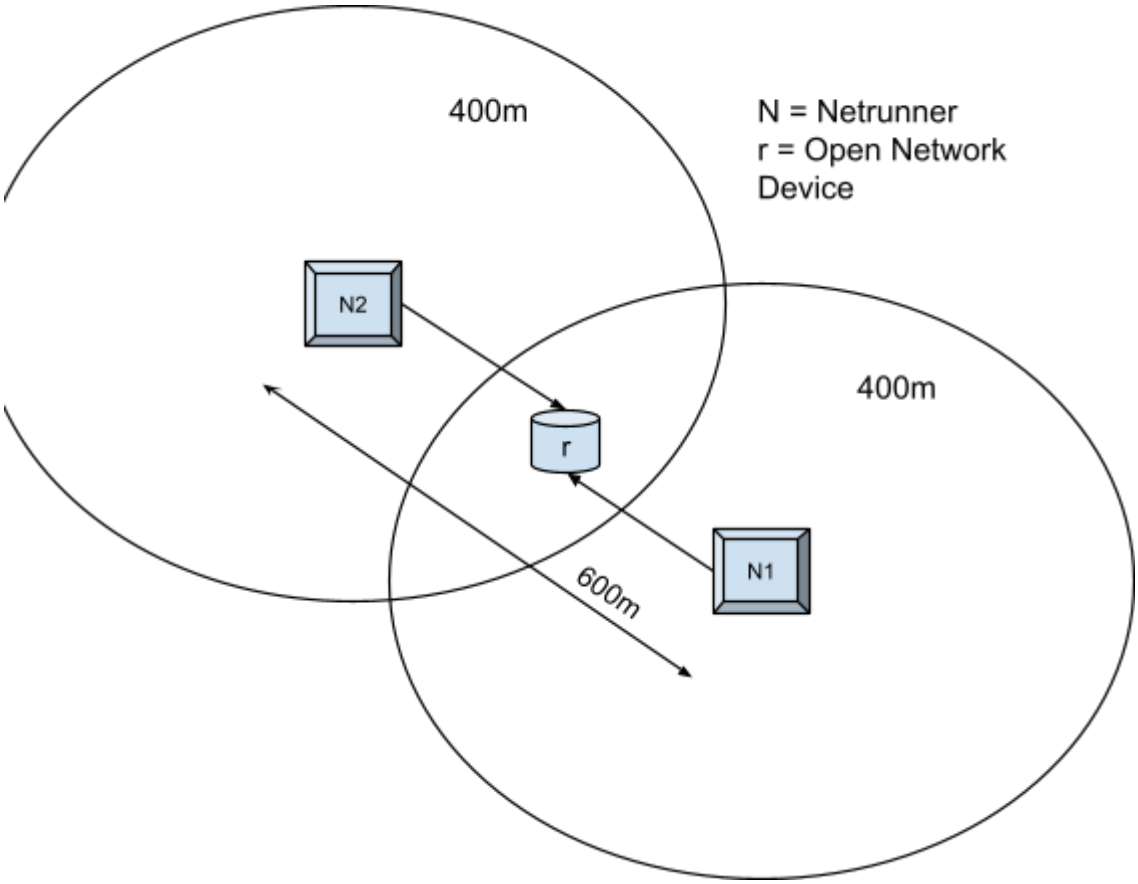
You may declare an action to be persistent - this is setting up something to run, and leaving it to run over any number of rounds. Persistent action can take one of three forms. An Action declared persistent by the player, an action that the GM rules will take more than one round to complete, and programs.

> Examples include:
> “I leave the Scanner running, looking for hackable stuff.” = [Locate//Modem] while this order is active, the Locate roll is applied to any detectable target, every turn, until stopped.
> “I conceal my connection for the duration.” = [Conceal//Modem] : while active, any attempt to back trace your connection must beat your conceal roll.
> “I Cipher the whole database.” = [Cipher//database] = The GM may rule a long duration task may take a number of combat rounds to process.
> “I Run the IC breaker to crack the systems security” = [Run Program//System]

[Mazu.Net] Every persistent action requires additional MU equal to (Command + Target) / 2. Programs do not require more MU than they already require to exist on your Deck.

Control Range [Mazu.Net]

Connections from a Netrunner contain vast amounts of data which can degrade over great distances when not operating strictly on the Net or via LDLs. Your standard range for broadcast is 400m. This range can ‘hop’ from open network points.



Computer Design [Mazu.Net]

Computers use the Command//Target system as well. **[More details needed.]**

Deck Design

[Mazu.Net] Use the Official Rules when building/looking at Deck Stats

Speed: When rolling Initiative on the Net, roll Computer's INT+1d10 OR Netrunner's REF + Speed + 1D10 vs Netrunner's REF + Speed + 1D10

Data Wall (or Data Gate): Otherwise known as IC or ICE. Used as a modifier in actions like SECURE. In any calculation where Data Wall is used, substitute it for a SECURE (or similar) command action.

MU: Used to store Programs or Persistent Scripts.

Programs

[Mazu.Net] Use the Official Rules when building/looking at Program stats.

Launching: To launch a program the user must make a Run program/User check. The DV of this check is equal to the program's strength times 4.

Multiple Instances: Unless otherwise noted a program can have an unlimited number of itself running at the same time, as long as it is being run by the registered user, **[Mazu.Net]** and you have enough MU.

Cyberware & Self-ICE [Mazu.Net]

Feeling fancy as you walk around with your NeuroComm? Got the latest Screamsheet thanks to your EntertainTrans? Checking your stock options as you Aero to the office with your NetlinkPro? **You're in danger.**

Protect yourself for whenever you hit the Net - Get ICE.

—

All Neuralware systems operate with an **Open Network** in order to make a constant connection with the Corporate Entity that owns the attached cyberware's licenses, make nightly calls to update firmware, deliver usage data, and more to help formulate the next best product. This leaves you vulnerable to attacks from those with the know-how, however.

Note: Neuralware comes equipped with basic Self-ICE.

—

Self-ICE is a type of cyberware that is intended to protect the software of your chrome.

Most systems come pre-installed with a form of basic Self-ICE, while better options for more security exist at a cost.

Effects: Self-ICE sets a base DIFF for intrusion over open-network, and then a modifier based on certain types of intrusion. Some Self-ICE includes anti-intrusion measures which immediately fire back at the infiltrator.

Note: Use the DIFF + connection method modifier as the 'attack' roll against any countermeasures the infiltrator may be using.

Products:

Neuralware Self-ICE, Basic

0eb (part of the purchase), DIFF 20 (-2 direct/+2 indirect), NONE

Biotechnica Neo Self-ICE

800eb, DIFF 20 (+2 direct/+2 indirect), NONE

Biotechnica "Guardian" Self-ICE

3,125eb, DIFF 20 (+5 direct/+5 indirect), BLACKOUT

Ardek Confidence MAX Self-ICE

1125eb, DIFF 20 (-2 direct/+4 indirect), DAGGER (1d4)

Self-ICE Math

Working numbers:

- Average Netrunner
    - Interface 5, Intelligence 5, and Strength 2-3 programs = 1d10+5+5+2.5 = 18
    - Self-ICE in this range = Cheap (200eb-800eb)

    - DIFF bottom = 20 + (+/-direct/indirect) would stop an Average Netrunner most of the time
- Great Netrunner
    - 8, 8, 6 = 1d10+8+8+6= 25.5
    - Self-ICE in this range = Expensive (1,950eb+)
- Top Tier Netrunner
    - 10, 10, 10 = 1d10+10+10+10 = 35.5
    - Self-ICE in this range = Military/Luxury (10,000eb+)

- Costs
    - Base cost
        - +1-4 DIFF: 200eb
        - +5 DIFF: 1,000eb
        - For every +1 DIFF above 5: +2,000eb (max +15)
    - Program Chips: Extra
    - +1 to DIFF (direct/indirect) = 150eb.
    - -1 to DIFF (direct/indirect) = -150eb.

Examples:

- A Self-ICE with DIFF 20 (+0/+10) and no programs would cost 10k (base) + 1,500 (+1 DIFF x10) = 11,500eb.
- A Self-ICE with DIFF 20 (+5/+5) and no programs would cost 1k (base) + 1,500 (+1 DIFF x10) = 2,500eb
- A Self-ICE with DIFF 20 (-3/+12) and no programs would cost 14k (base) + 1,800 (+1 DIFF x12) - 450 (-1 DIFF x3) = 15,350eb

Programs:

Programs come as an additional cost to the Self-ICE, are limit 1, and are typically unique or a special version of an existing program. There exists unique, after-market add-ons that allow for more variety, however.

Effect: When Self-ICE contains a program and it detects an unauthorized attempt at logging into the network, sub-network, or architecture, it activates the contained program targeting the intruder.

Note: Self-ICE uses its DIFF + appropriate direct/indirect modifier as its DETECT, should the intruder be concealed.

—

Note on custom programs: Take a base program and use 10% of the cost. Try to

reduce all numbers to about 2/3rds or 66% effectiveness. SWORD is 6250eb, Strength 3, and 1d6. DAGGER is 625eb, Strength 2, 1d4.

NAME
Cost: eb
Use:

ICETRAP
Cost: 650eb
Use: Locks the invading Netrunner in place by looping his data-feed for 1d7 turns (4 turns is enough to get a good trace on their location in realspace.)

DAGGER
Cost: 625eb
Strength: 2
Use: Causes a lethal energy feedback to the invading Netrunner, dealing 1d4 physical damage.

BLACKOUT
Cost: 625eb
Use: Delivers a powerful modulated shock that knocks the invading Netrunner out for 1d4 hours.

Expansion Chip
Cost: 1,000eb per MU (max 4)
Use: Expansion chips are special, after-market installations made popular by edgerunners with a taste for dangerous encounters.
Expansion Chips come in 1, 2, 3, and 4 MU versions, and allow the user to have a Program installed on them for their Self-ICE to deploy.
Note: Program cost is separate.