

Scepter

User Help Documentation

Version 1.0.7

Table of Contents

- Main Window
- Glossary
- Getting Started
  - Example 1 — Simple: Small Emergency Radio Net
  - Example 2 — Medium: Community Emergency Response
  - Example 3 — Complex: Full Operational CEOI
  - Overview
  - Generated Code Patterns
  - Generation Modes
  - CEOI Defaults
    - Cover Settings
    - Notes
    - Options
    - Unit Options
    - Group Options
    - Word List Options
    - Documents Options
    - Location Options
    - Frequency Options
    - DRYAD Options
    - SARNEG Options
    - Challenge/Response Options
    - OTP Options
    - SOI Options
  - Units
    - Unit Editor
      - Details
      - Groups
      - Notes
      - Callsigns

- Authentication Codes
- Passwords
- Overrides
- Generated Codes
- Assets
- Unit Selector

Groups

- Group Editor
  - Members
  - Member Of
  - Options
  - Notes
  - Generated Codes
  - Assets
- Group Selector

Word List

- Word Editor
  - Edit
  - Generated Codes
  - Notes
- Word Edit Panel
- Word List Encoder/Decoder

Documents

- Document Editor
  - Edit
  - Notes
  - Generated Codes
- Document Edit Panel

Locations

- Location Editor
  - Edit
  - Notes

- Generated Codes
- Location Edit Panel
- Frequencies
- Frequency Editor
- Edit
- Notes
- Generated Codes
- Frequency Edit Panel
- Bands
- Band Editor
- Create Frequency List
- DRYADs
- DRYAD Panel
- DRYAD Viewer
- Authenticate with DRYAD
- Encrypt Numbers with DRYAD
- Decrypt Numbers with DRYAD
- SARNEGs
- SARNEG Panel
- SARNEG Viewer
- Challenge/Response
- CRD Words Viewer
- One Time Pads
- OTP Table Panel
- OTP Viewer
- OTP Import
- OTP Message Encrypt
- OTP Message Decrypt
- OTP Hash
- OTP Message
- Reverse OTP Message
- Signals Operating Instructions (SOI)

- SOI Editor
  - Details
  - Units
  - PACE
  - Comm Schedule
  - Notes
  - Assignments
  - Assets
- Comm Window Editor
- Comm Schedule Generator
- Asset Assignments
- Asset Print
- Group Asset Print
- Entity Selector
- Coordinate Conversion
- QR Code Generator
- QR Code Decoder
- Camera Capture
- Print Jobs
- Print Configurations
  - Print Configuration Editor
- Print Templates
- Report Data Sources
  - Data Source Script Editor
  - Writing Custom Data Sources
- Exporting Data
  - Periods and Divisions
  - Export Scripts
  - Export Script Editor
  - Writing Export Scripts
  - Units (CSV)
  - Units (JSON)

- Unit Codes (CSV)

Reports

- Basic Text Cover Sheet
- Cover Sheet (pdf)
- Authentication Codes
- Call Signs
- Passwords
- Daily Codes (Multi Divisions Per Page)
- Daily Codes (Single Division Per Page)
- Daily Codes Condensed
- Unit Detail
- Group Roster
- Document Codes - Encrypt
- Document Codes - Decrypt
- Frequency Codes - Encrypt
- Frequency Codes - Decrypt
- Location Codes - Encrypt
- Location Codes - Decrypt
- Wordlist Encode
- Wordlist Decode
- Challenge Response Words
- DRYADs
- SARNEGs
- SOI
- One Time Pad
- One Time Pad IN
- One Time Pad IN Simplified
- One Time Pad OUT
- One Time Pad OUT Simplified
- OTP DIANA Conversion Table
- OTP Diana Plus Conversion Table
- OTP Numeric Conversion Table

- Month Calendar
- Data Source Reference
- WORDLIST_ENCODE / WORDLIST_DECODE
- Division-Column Code Grids
- UNIT_DAILY_CODES
- UNIT_CODES_CONDENSED
- UNITS (Unit Detail)
- GROUPS
- CEOI
- Location / Document / Frequency Decrypt
- CALENDAR
- CYPHERTABLES (DRYAD)
- SARNEG
- CHALLENGE_RESPONSE
- ONE_TIME_PAD
- OTP Codex Tables
- SOI
- CEOI (Root Object)
- Unit
- Group
- Location
- Frequency
- Document
- Word
- DRYAD
- Sarneg
- ChallengeResponse
- OneTimePad & Codex
- SOI

Settings

- Random Number Generation

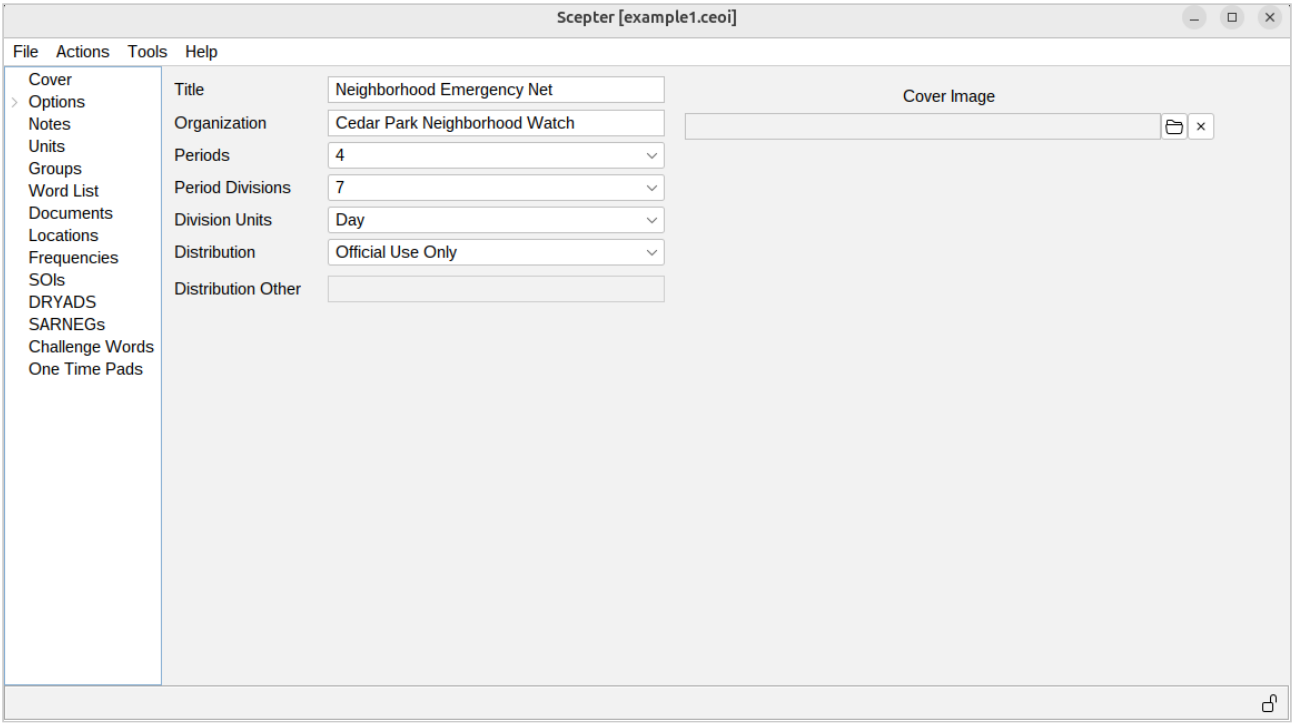
License

Clear Crypto

Set File Password

Enter Password

Main Window



The Main Window is the primary interface for working with CEOI files in Scepter. It is divided into three main areas: the menu bar along the top, a navigation tree on the left, and a content area on the right that displays the currently selected panel.

Window Title Bar

The title bar displays the Scepter version number and the path to the currently open CEOI file. If no file has been saved yet, it shows `[*** New ***]`. In demo mode (unlicensed), the title bar includes the word `DEMO`. If the installed license was issued for a different computer, it says `Invalid License` instead; see License.

Navigation Tree

The left side of the window contains a tree menu that you use to navigate between panels. Selecting an item in the tree switches the content area on the right to show the corresponding panel. The tree contains the following top-level sections:

- **Cover** – Basic CEOI information such as title, organization, periods, and classification. See Cover Settings.
- **Options** – Generation settings for each asset type. Expanding this node reveals sub-items for Units, Groups, Word List, Documents, Locations, Frequencies, DRYADS, SARNEGs, Challenge/Response, One Time Pads, and SOIs. See Options.
- **Notes** – Free-text notes associated with the CEOI. See Notes.
- **Units** – The list of units included in the CEOI. See Units.
- **Groups** – Net groups composed of units. See Groups.

- **Word List** – Codewords used for encoding messages. See Word List.
- **Documents** – Document code assignments. See Documents.
- **Locations** – Location code assignments. See Locations.
- **Frequencies** – Radio frequency assignments. See Frequencies.
- **SOIs** – Signal Operating Instructions. See SOIs.
- **DRYADS** – DRYADs. See DRYADS.
- **SARNEGs** – SARNEG tables. See SARNEGs.
- **Challenge Words** – Challenge and response password pairs. See Challenge Words.
- **One Time Pads** – OTP sheets for message encryption. See One Time Pads.

> When a panel supports importing or exporting list data, the **Import List** and **Export List** items in the Tools menu become enabled for that panel.

Menu Bar

File Menu

Menu Item	Shortcut	Description
Open...	`Ctrl`+`O`	Open an existing CEOI file.
Open as Template...	`Ctrl`+`Shift`+`O`	Open an existing CEOI file as a template for a new CEOI. The file is loaded but treated as a new unsaved document, allowing you to reuse its structure without modifying the original.
New	`Ctrl`+`N`	Create a new blank CEOI. If the current CEOI has unsaved changes, you are prompted to save first.
Save	`Ctrl`+`S`	Save the current CEOI. If the file has not been saved before, you are prompted to choose a location and filename. Requires a license; with an invalid license the license violation notice is shown for one minute first.
Save As...		Save the current CEOI to a new file. Requires a license.
Recent Files...		A submenu listing recently opened CEOI files for quick access. This submenu is visible only when the recent files list is enabled in Preferences.
Print...		Open the Print Job dialog to print the current CEOI.
Export...		Run an export script over the whole CEOI and save what it writes — a spreadsheet of units, a JSON file — to a file you choose. See Exporting Data. Not the same as Tools > Export List, which writes the table on screen.

Set Password...		Set or change the encryption password for the current CEOI file. See Set File Password.
Settings submenu:		
Print Jobs...		Configure print job settings. See Print Configuration.
Print Templates...		Manage print templates. See Print Templates.
Report Data Sources...		Read, copy and write the scripts that supply report data. See Report Data Sources.
Export Scripts...		Read, copy and write the scripts behind File > Export. See Export Scripts.
CEOI Default Settings...		Edit the default values used when creating a new CEOI. See CEOI Defaults.
Preferences...		Open the application preferences dialog. See Preferences.
Set Current as Default		Save the currently open file as the default file to open automatically at startup. The next time Scepter launches, it will open this file instead of starting with a blank CEOI.
Quit	`Ctrl`+`Q`	Exit Scepter. If the current CEOI has unsaved changes, you are prompted to save first.

Actions Menu

Menu Item	Description
Generate Crypto...	Generate all cryptographic material (callsigns, authentication codes, DRYADs, SARNEGs, OTPs, etc.) for the current CEOI based on the configured options. A progress dialog tracks the generation process.
Remove Generated Crypto...	Remove all previously generated cryptographic material from the current CEOI.
Reset To Default Values	Reset all CEOI settings and options back to their default values. You are asked to confirm before proceeding.

> Resetting to defaults overwrites all General and Settings data. Generated crypto is not affected, but the settings that produced it will be replaced.

Tools Menu

Menu Item	Description
Convert Coordinates...	Open the coordinate converter utility to convert between coordinate formats.
Create QR Code...	Generate a QR code from text. See Create QR Code.
Decode QR Code...	Decode a QR code from an image file. See Decode QR Code.
Encode/Decode with Wordlist...	Encode or decode a message using the CEOI's word list. Only enabled when the CEOI contains words. See Word List Encode/ Decode.
Encrypt with OTP...	Encrypt a message using a one time pad. Only enabled when the CEOI contains OTPs.
Decrypt with OTP...	Decrypt a message using a one time pad. Only enabled when the CEOI contains OTPs.
Reverse Encrypt OTP...	Perform a reverse OTP encryption. See Reverse OTP Encrypt.
Hash Message...	Hash a message using the OTP hash utility. See Hash Message.
Authenticate with DRYAD...	Challenge a station, or answer a challenge, with a DRYAD. See Authenticate with DRYAD.
Encrypt Numbers with DRYAD...	Encrypt a short number with a DRYAD. See Encrypt Numbers with DRYAD.
Decrypt Numbers with DRYAD...	Decrypt a number encrypted with a DRYAD. See Decrypt Numbers with DRYAD.
Import List...	Import list data into the currently selected panel (enabled only for panels that support it).
Export List...	Export list data from the currently selected panel (enabled only for panels that support it). To export from the whole CEOI with an export script, use File > Export instead; see Exporting Data.
Import One Time Pad...	Import an OTP from an external file. See OTP Import.
Import words for SARNEGs...	Import a text file of words to use when generating SARNEG tables.
Import words for Challenge/Response Passwords...	Import a text file of words to use when generating challenge and response password pairs.

Help Menu

Menu Item	Description

Help Table of Contents...	Open the help system table of contents.
Save log files	Save the application log files to a zip archive in your documents directory for troubleshooting.
Install Update...	Install a program update from a downloaded update file. You must save any unsaved changes first.
License...	Request a license, or install one that has been sent to you. Shown only when this copy is unlicensed or its license was issued for a different computer. See License.
About...	Display the version, and the license ID of this installation with a button to copy it.

Status Bar

The bottom of the window contains a status bar. It displays a lock icon indicating whether the current CEOI file is encrypted (locked) or unencrypted (unlocked). You can set or change the file password from **File > Set Password...**.

Unsaved Changes

When you create a new CEOI, open a file, or quit the application, Scepter checks whether the current CEOI has unsaved changes. If changes are detected, you are prompted to save, discard, or cancel the operation.

> CEOI files use the `.ceoi` file extension.

Glossary

Definitions for terms used throughout Scepter and its help documentation, listed alphabetically.

Term	Definition
Actor	Any entity that can hold assigned crypto assets — specifically a Unit or a Group. When an asset (OTP, DRYAD, etc.) is assigned to an actor, it appears in that actor's Assets tab.
Asset	A piece of cryptographic or operational material that can be assigned to units or groups. Asset types are: DRYAD, One-Time Pad, SARNEG, Challenge/Response, and SOI.
Asset Assignment	The association between a specific asset (DRYAD, OTP, SARNEG, etc.) and one or more units or groups. Assignments are managed through the Asset Assignment Dialog and are reflected in the actor's Assets tab and on printed output.
Authentication Code	A short code assigned to a unit or group that allows a radio operator to verify the identity of a station during a net. Authentication codes are generated per unit and per group and are listed in the Auth Codes tab of the unit editor and the Codes tab of the group editor.
Callsign	A short identifier assigned to a unit or group that is used on the radio net in place of the true name or designation, providing a basic level of identity protection. Callsigns are generated per unit and per group and are listed in the Callsigns tab of the unit editor and the Codes tab of the group editor.
CEOI	*Communications-Electronics Operating Instructions.* The master document produced by Scepter that bundles all communications security material — call signs, frequencies, DRYADs, OTPs, passwords, and more — for distribution to units in a command. Historically also called SOI (Signal Operating Instructions). See What is a CEOI?
Challenge Word	The first half of a challenge/response pair. One party issues this word; the other must reply with the correct **Response Word** to prove their identity. See also: **Duress Word**.
Challenge/ Response	A system of paired passwords used for identity verification over radio. One party challenges with the challenge word; the correct response word confirms identity. A third **duress word** signals that the responder is under coercion. See Challenge/Response.
Code Pattern	A template string made of `L` (letter) and `N` (digit) characters that controls the format of generated codes. For example, `LNL` produces codes like `K3T`. See Generated Code Patterns.
Code Space	

	The total number of distinct codes that a given **code pattern** can produce. If the code space is smaller than the number of items that need codes, generation will fail with an error. See Generated Code Patterns.
Codex	The cipher alphabet used by DIANA-type one-time pads. A codex maps each letter of the alphabet to a digit (and vice versa), enabling letter-based OTP encryption and decryption. The DIANA and DIANA+ codex tables are included in printed OTP output.
Comm Schedule	The list of scheduled communication windows within a Signals Operating Instruction (SOI). Each entry specifies a time window, frequency, and other parameters. See Comm Schedule and the Comm Schedule Generator.
Comm Window	A single scheduled contact entry within a **comm schedule**. Specifies the time, duration, frequency, and any associated prowords for a planned radio contact. Edited in the Comm Window Editor.
DIANA	A letter-based one-time pad format. DIANA OTPs contain random letters rather than digits and use a codex table to convert between letters and numbers. **DIANA+** is a variant that includes both letters and numbers. See One Time Pads and OTP Encrypt.
Division	A time subdivision within a CEOI **period**. A period may be divided into multiple divisions (days, shifts, or other intervals), each with its own crypto material. The label used for divisions (e.g., "Day", "Shift") is configured in Cover Settings.
Document	A reference document (orders, maps, overlays, etc.) that is assigned a coded identifier in the CEOI. During transmission, operators use the code instead of the real document name. Managed in Documents.
DRYAD	A numeric table used to encrypt numbers — such as grid coordinates, frequencies, or times — into letter groups for secure voice or written transmission. Each DRYAD covers a single period/division. See DRYADs and the DRYAD Viewer.
Duress Word	A special response word within a challenge/response pair that signals to friendly forces that the responder is operating under coercion. Using the duress word instead of the normal response word alerts the challenger without alerting the adversary.
Frequency	A radio frequency assignment in the CEOI. Frequencies are assigned coded identifiers and linked to units, groups, and SOIs. Managed in Frequencies.
Generation Mode	Controls whether generated codes (callsigns, auth codes, etc.) are unique across the entire CEOI, unique per period, or unique per division. See Generation Modes.
Group	

	A named collection of units that communicates as a net. Groups can be members of other groups. Like units, groups receive generated codes and can be assigned crypto assets. Managed in Groups.
Location	A geographic position (expressed as coordinates, MGRS, UTM, Maidenhead, What3Words, or BIP39) assigned a coded identifier for secure reference during operations. Managed in Locations.
Message Hash	A cryptographic hash of a message, optionally salted with an OTP or DRYAD, used to verify that a message was not altered in transit. Both sender and receiver hash the same message with the same salt; matching hashes confirm integrity. See Message Hash Tool.
One-Time Pad (OTP)	A random key used for encrypting or decrypting messages. Each pad is used exactly once and then destroyed. OTPs in Scepter come in three types: **Numbers** (digit-only), **DIANA** (letter-based), and **DIANA+** (letters and numbers). See One Time Pads, OTP Encrypt, and OTP Decrypt.
PACE	*Primary, Alternate, Contingency, Emergency.* A communications planning framework that designates four escalating means of contact in case the primary method fails. PACE plans are configured per SOI in the PACE tab of the SOI editor.
Password	A generated code assigned to a unit or group used for access control or identity verification purposes beyond authentication codes. Passwords are generated per unit and per group and are listed in the Passwords tab of the unit editor and the Codes tab of the group editor.
Period	A top-level time window in the CEOI during which a specific set of crypto material is valid. A CEOI may cover multiple periods (e.g., daily, weekly). Each period may be further divided into **divisions**. Periods are configured in Cover Settings.
Print Template	A reusable layout definition that controls how CEOI data is formatted on printed pages. Templates are managed in Print Templates and applied through Print Configurations.
PROWORD	*Procedure word.* A standardized word or phrase with a specific meaning on a radio net (e.g., "ROGER", "WILCO", "BREAK"). In Scepter, prowords are associated with SOI frequency assignments to indicate how each net is to be called. Configured in the SOI Details tab.
QR Code	A two-dimensional barcode that encodes text or data for machine scanning. Scepter can generate QR codes containing DRYAD or OTP data for rapid distribution and can decode QR code images. See QR Code Generator and QR Code Decoder.
Response Word	

	The correct reply to a **challenge word** in a challenge/response pair. Giving the correct response word proves identity to the challenger. See also: **Duress Word**.
SARNEG	*Search and Rescue Numerical Encryption Grid.* A grid-based numeric encoding table used to encode numerical values during search and rescue operations. Similar in purpose to DRYAD but structured differently. See SARNEGs and the SARNEG Viewer.
SOI	*Signals Operating Instructions.* A self-contained communications plan for a specific operation or time window. An SOI bundles together frequencies (primary, alternate, contingency, emergency), a PACE plan, a comm schedule, and assigned units. Multiple SOIs can exist within a single CEOI. See SOIs and the SOI Editor.
Unit	An individual station, element, or operator within the command. Units receive generated codes (callsigns, auth codes, passwords), can be members of groups, and can be assigned crypto assets. Managed in Units and edited in the Unit Editor.
Word List	A collection of codewords assigned encoded identifiers for use in message encoding. Operators substitute codewords for plaintext concepts during transmission. The word list can also be used to encode and decode messages via the Word List Encoder/Decoder. Managed in Word List.

Getting Started: Creating a CEOI

A **Communications-Electronics Operating Instructions** (CEOI) document is a pre-planned, printed reference that gives every member of a communications network everything they need to operate securely: who they are on the air, where to transmit, how to prove their identity, and how to protect sensitive information. Scepter generates all of that material in a single file, prints it into individual operator cards, and lets you update it on a regular schedule so old codes never stay in circulation longer than necessary.

Why Create a CEOI?

Communications security is not only a military concern. Any group that operates radios in an emergency — neighborhood emergency response teams, amateur radio operators supporting public events, backcountry expedition teams, or preparedness organizations — benefits from pre-planned call signs, frequencies, and authentication procedures for the same reasons the military does:

- **Clarity under stress.** In an emergency there is no time to improvise call signs or argue about which frequency to use. A CEOI settles all of that in advance.
- **Authentication.** Radio signals are anonymous. Challenge/response words and authentication codes let operators confirm they are talking to a friendly station and not an adversary mimicking one.
- **Compartmentation.** Not everyone needs all the codes. Scepter lets you print tailored code cards for each unit so that the capture of one card does not expose the entire network.
- **Rotation.** Codes that change daily or weekly limit the damage if a card is lost. A CEOI covers up to 10 periods of up to 10 divisions each so you can issue a full block of rotating codes at once and retire each set automatically.
- **Offline availability.** A printed CEOI works when phones, internet, and computers are all offline — exactly when you need it most.

Key Concepts

Understanding these terms will help you configure Scepter correctly.

Periods and Divisions

A CEOI covers a span of time divided into **periods** (the largest unit, e.g., a month or a quarter) and **divisions** within each period (the smallest unit, e.g., a day or a shift). Codes such as call signs and challenge words are generated for each period–division combination, so they change on a schedule. A simple group might use 1 period of 7 daily divisions (one week of daily codes). Scepter supports a maximum of 10 periods with up to 10 divisions each (100 total code-change slots).

Units and Groups

A **unit** is a single station or operator. A **group** is a named collection of units. Call signs, authentication codes, and passwords are generated per unit–group pair, so the same operator can have different call signs in different nets. Groups can contain sub-groups for hierarchical organization.

Call Signs

Temporary radio identifiers that change each division. Scepter generates them from a pattern you define (e.g., two random letters followed by two digits) and assigns a unique value to each unit for each period and division.

Authentication Codes

Short numeric or alphanumeric codes used to verify a station's identity during a net check-in. Different from challenge/response words: authentication codes are typically announced by the station identifying itself, while challenge/response is a back-and-forth exchange.

Challenge / Response Words

A pair of plain-language words. One station issues the *challenge* word; the other must reply with the correct *response* word to prove it is a friendly. A third *duress* word signals that the responder is under coercion. These change each division.

DRYADs

A printed grid used to encrypt short numeric messages such as grid coordinates. The operator looks up a row letter and column digit to find the ciphertext character. DRYADs require no electronics and are fast under field conditions. Scepter generates a new table for each period or division according to your configuration.

SARNEGs

A 10-character keyword where each letter substitutes for a digit 0–9. Used in search-and-rescue and similar operations to encode numeric values in voice transmissions. The keyword changes each division.

One-Time Pads (OTPs)

Random character sequences used for pencil-and-paper encryption of text messages. Each pad is used exactly once, then destroyed. OTPs provide information-theoretic security: even with unlimited computing power, an encrypted message cannot be broken without the pad. Scepter generates and prints OTPs in the Numbers, DIANA, or DIANA+ formats.

Signals Operating Instructions (SOI)

A structured net description attached to a CEOI: it defines the PACE (Primary–Alternate–Contingency–Emergency) frequency plan, the units assigned to the net, a communication schedule, and the SARNEG for the net. An SOI is what a radio operator carries to know how to operate a specific net.

Frequencies, Locations, and Documents

You can define radio frequencies, geographic locations, and documents in Scepter and generate short codes for each one so that operators can reference them in messages without speaking the actual frequency or location aloud.

Workflow Overview

1. **Cover & Options** — Set the title, organization, distribution, number of periods and divisions, and a count for each feature you need. A count of 0 leaves a feature out; there is no separate enable switch.
2. **Units** — Add each operator or station.
3. **Groups** — Organize units into nets; configure call sign and code generation options per group.
4. **Frequencies, Locations, Documents** — Add the operational data that units will reference during operations.
5. **Crypto Assets** — Add or generate DRYADs, SARNEGs, OTPs, and challenge/response words as needed.
6. **SOIs** — Define each net with its PACE plan and comm schedule.
7. **Generate & Print** — Run **Actions** → **Generate Crypto...**, then define print jobs under **File** → **Settings** → **Print Jobs...** and run them with **File** → **Print...** to produce individual operator packets.

Examples

The following step-by-step examples cover three common scenarios at increasing levels of complexity:

- Example 1 — Simple: Small Emergency Radio Net
  Six operators, one group, four weekly periods of daily call signs and challenge words. The minimum viable CEOI for a neighborhood or event communications team.
- Example 2 — Medium: Community Emergency Response Organization
  Fifteen operators in two groups, two periods of ten days each, authentication codes, a daily DRYAD, and an SOI with a PACE plan for each of the two nets.
- Example 3 — Complex: Full Operational CEOI
  Forty operators across five code-issuing groups, the maximum 10 × 10 period/division configuration, the full crypto suite (DRYADs, SARNEGs, one-time pads), three SOIs, and coded frequency, location and document references.

Related Topics

- What is a CEOI?
- Generation Modes
- Generated Code Patterns
- Cover Settings
- Units
- Groups
- Print Jobs

Example 1 — Simple: Small Emergency Radio Net

A neighborhood emergency preparedness group has six licensed amateur radio operators who want to be able to communicate securely during a disaster. They meet monthly, all know each other, and want a simple card they can keep in their go-bags that tells them their call sign for the week and the challenge word for each day. They do not need encryption: just organized, authenticated communications.

The finished CEOI ships with Scepter as `examples/example1.ceoi`. Every value below is the value in that file, so you can open it and compare as you read.

Scenario

Setting	Value	Why
Operators	6 (net control + 5 field)	Small enough to fit on a single card
Periods	4	One monthly issuance covering four weeks
Period Divisions	7	Daily code rotation through the week
Division Units	Day	Each division is one calendar day
Call signs	1 per unit per day	Keeps real call signs off the air
Challenge / Response	1 set per day	Identity verification on net check-in
Authentication codes	None	Too much to carry for a group this size
Passwords	None	No equipment or secondary authentication
DRYADs, SARNEGs, OTPs	None	No encrypted traffic
Frequencies	3	Repeater, simplex backup, county emergency
Locations, Documents	None	Nothing to reference by code

Step 1 — Cover

Create a new CEOI (**File** → **New**) and select **Cover** in the navigation tree on the left. Fill in:

- **Title:** Neighborhood Emergency Net
- **Organization:** Cedar Park Neighborhood Watch
- **Periods:** 4
- **Period Divisions:** 7
- **Division Units:** Day
- **Distribution:** Official Use Only (or Training for a practice run)

Step 2 — Options

Select **Options** in the tree. It expands into one page per feature. There is no enable/disable switch anywhere in Scepter: a **Count** of 0 is how you turn a feature off, and a count of 1 or more is how you turn it on.

- **Word List, Documents, Locations, Frequencies:** Count 0. Nothing in this CEOI is referenced by code.
- **DRYADS, SARNEGs, One Time Pads:** Count 0.
- **Challenge/Response:** Generation Mode **Division**, Count **1** — one challenge, response and duress word for each day.

> **Add to Word List** and **Count** are separate controls and do different things. Count decides whether an entity gets a generated code. Add to Word List decides whether the entity also appears in the CEOI word list, so it can be encoded and decoded with **Tools → Encode/Decode with Wordlist**. This example leaves both off for frequencies.

Step 3 — Frequencies

Select **Frequencies** in the tree and add three entries. They are printed as a plain reference list; no codes are generated for them.

Name	Frequency	Mode	Notes
Primary	146.940	FM	Cedar Park repeater, 100.0 Hz tone
Alternate	146.520	FM	National simplex calling frequency
Emergency	155.160	FM	County emergency services

Step 4 — Units

Select **Units** and add one unit per operator. Only two fields matter here:

- **Name:** the operator's name and amateur call sign, for example `Jane Smith (W1ABC)`
- **Number:** 1 through 6, with net control as number 1

Leave Field A, Field B and Field C empty; they are for organizations that need to record a role or an element on each unit, which this group does not.

Step 5 — Group and Call Signs

Select **Groups** and add one group named **Emergency Net**. On the **Members** tab add all six units, then open the **Code Options** tab:

- **Generate Group Codes:** off — the group is not itself a station on the air, so it needs no call sign of its own
- **Generate Group Member Codes:** on
- **Callsign Pattern:** `LLNN`

- **Callsign Count:** 1
- **Authentication Code Count:** 0
- **Password Count:** 0

> Counts on this tab are per unit *per division*. A Callsign Count of 1 across 4 periods of 7 divisions gives each operator 28 call signs — one for every day the CEOI covers. Patterns are built from `L` (random letter A–Z) and `N` (random digit 0–9), and those are the only characters allowed. `LLNN` produces call signs like `BK47` or `ZR02`. See Generated Code Patterns for the full rules.

Step 6 — Generate

Select **Actions** → **Generate Crypto...**. Scepter fills in every call sign and every challenge/response set in one pass. The challenge, response and duress words are drawn from the word list that ships with Scepter, so there is nothing to type.

Generation is safe to repeat: codes that already exist are kept, and only what is missing is created. To replace material rather than add to it, use **Actions** → **Remove Generated Crypto...** first.

> Brief every operator on the duress procedure. If a station answers a challenge with the duress word instead of the response word, it is signaling that the operator is under coercion. Treat it like a mayday.

Step 7 — Print

Reports are produced by print jobs. Define them under **File** → **Settings** → **Print Jobs...**, then run one with **File** → **Print...**. For this CEOI two reports carry everything the operators need:

- Call Signs — every operator's call sign for each day of each period
- Challenge Response Words — the challenge, response and duress words for each day

You can also print straight from a list: select the operators on the **Units** page and press **Print**, which offers only the reports that suit a unit. Print one copy per operator, laminate if you can, and hand them out at the next meeting.

Result

Each operator leaves with a card holding four weeks of daily call signs and the challenge/response words for each day. Every card is identical except for the call signs, which are unique to that operator. When a period ends, shred the cards and issue the next set; when all four periods are used, generate a new CEOI for the following month.

Next Steps

When the group is comfortable with this CEOI, consider adding:

- Authentication codes, so operators prove identity at check-in
- A DRYAD, so grid coordinates can be passed securely by voice
- A formal SOI defining the PACE plan for the net

Example 2 adds all three.

Related Topics

- Getting Started Overview
- Example 2 — Medium
- Cover Settings
- Options
- Units
- Groups
- Challenge Words
- Generated Code Patterns
- Call Signs Report
- Challenge Response Words Report

Example 2 — Medium: Community Emergency Response Organization

A community emergency response organization has fifteen operators split into two functional teams: a Leadership group of five and a Field Operations group of ten. They issue a CEOI covering two operational periods of ten days each, so codes rotate daily. On top of call signs and challenge words they want authentication codes for net check-in, a DRYAD so grid coordinates can be passed by voice, and a formal SOI for each of their two nets.

The finished CEOI ships with Scepter as `examples/example2.ceoi`. Every value below is the value in that file.

Scenario

Setting	Value	Why
Operators	15 across 2 groups	Leadership and Field Operations
Periods	2	Two operational periods per issuance
Period Divisions	10	Ten days per period; codes rotate daily
Division Units	Day	Each division is one calendar day
Call signs	1 per unit per day	Standard identification on the air
Authentication codes	1 per unit per day	Identity verification on check-in
Challenge / Response	2 sets per day	One per net
DRYADs	1 per day	Encrypted grid coordinate transmission
SARNEGs, OTPs, Passwords	None	Added in Example 3
Frequencies	6, coded once per period	Command net and field net, plus backups
Locations	4, coded once per period	EOC, two staging areas, hospital
SOIs	2	Command Net and Field Operations Net

Step 1 — Cover

Create a new CEOI and fill in the **Cover** page:

- **Title:** Community Emergency Response — CEOI
- **Organization:** Riverside Community Emergency Response Team
- **Periods:** 2

- **Period Divisions:** 10
- **Division Units:** Day
- **Distribution:** Official Use Only

Step 2 — Options

Work down the **Options** branch of the tree. A **Count** of 0 leaves a feature out; anything higher turns it on.

Options page	Settings
Units	**Field A Label** = Role. Field B and Field C are unused.
Frequencies	Generation Mode **Period**, Pattern `LLN`, Count **1**, Add to Word List **on**
Locations	Generation Mode **Period**, Pattern `LLN`, Count **1**, Add to Word List **on**, Report Detail Field **MGRS**
DRYADS	Generation Mode **Division**, Count **1** — a new table each day
Challenge/Response	Generation Mode **Division**, Count **2** — one set per net per day
Word List, Documents, SARNEGs, One Time Pads	Count **0**

> Generation Mode decides how often a code changes, and it is separate from Count. **Period** gives each frequency one code that holds for the whole period; **Division** would give it a new code every day. See Generation Modes.

Step 3 — Frequencies

Six entries on the **Frequencies** page, organized by net:

Name	Frequency	Mode	Notes
CMD Primary	147.180	FM	Command net primary, 103.5 Hz tone
CMD Alternate	147.480	FM	Command net simplex alternate
CMD Emergency	155.160	FM	County emergency services
OPS Primary	146.550	FM	Field operations primary simplex
OPS Alternate	146.580	FM	Field operations alternate simplex
ADMIN	446.000	FM	Administrative, non-tactical traffic

Every frequency picks up its code count from Options. A single frequency can override that on the **Codes** tab of its editor, which this example does not use.

Step 4 — Locations

Four sites on the **Locations** page. Scepter stores every coordinate format at once, so entering latitude/longitude and MGRS costs nothing and lets operators use whichever they prefer.

- Emergency Operations Center — `10T ER 12345 67890`
- Primary Staging Area — `10T ER 13120 66040`
- Alternate Staging Area — `10T ER 10880 69210`
- Regional Hospital — `10T ER 14260 68350`

With Add to Word List on, each location also lands in the CEOI word list, so it can be encoded and decoded with **Tools → Encode/Decode with Wordlist**.

Step 5 — Units

Fifteen units on the **Units** page. Numbers run 1–5 for Leadership and 6–15 for Field Operations, and the **Role** field (Field A, renamed in Options) carries the position: Incident Commander, Deputy IC, Operations Chief, Planning Chief, Logistics Chief, Field Team Leader, Field Operator.

Step 6 — Groups

Two groups on the **Groups** page. The **Code Options** tab of each one decides how its members' call signs and authentication codes are built. Both groups have **Generate Group Codes** off and **Generate Group Member Codes** on.

Group	Callsign Pattern	Callsign Count	Authentication Code Pattern	Authentication Code Count
Leadership Group (5)	`LN`	1	`NNNN`	1
Field Operations Group (10 + the Operations Chief)	`LLNN`	1	`NNNN`	1

The short `LN` pattern makes a Leadership call sign recognizable against the longer field call signs the moment it is heard.

> A unit can belong to more than one group and is issued a separate call sign in each, which is correct: its identity depends on the net it is working. In this CEOI the Operations Chief sits in Leadership and also runs the field net, so they are a member of both groups and carry two call signs for every day.

> Pattern space is checked before anything is generated. Scepter needs a pattern with room for at least 1.5 times the codes being asked of it, counting every unit, period and division. Leadership needs 5 × 2 × 10 = 100 call signs and `LN` holds 260, so it fits. Widen the pattern if Scepter reports insufficient code space.

Step 7 — Generate

Actions → Generate Crypto... produces everything at once: call signs and authentication codes for both groups, the frequency and location codes, the twenty DRYAD tables (one per day of each period), and the forty challenge/response sets.

DRYAD tables are random and need no data entry. To use one, pick a row letter and trace across to the column under each digit of your message; the letter at the intersection is the ciphertext character. The receiver reverses the lookup on the same table.

> Brief every operator on DRYAD procedure before the CEOI is issued. A field exercise is the best way to confirm that everyone can actually build and read a message under pressure.

Step 8 — SOIs

Two SOIs on the **SOIs** page. Each is edited across several tabs: **SOI** for the title, objective, location, control unit and comm schedule length; **PACE** for the four-tier frequency plan; **Units** for who is on the net; **Comm Schedule** for the daily windows; and **Assets** for the material issued to the net.

Command Net SOI

- **Objective:** Coordination between Leadership and team leaders
- **Location:** Emergency Operations Center
- **PACE:** CMD Primary / PROWORD STEEL · CMD Alternate / IRON · CMD Emergency / COPPER · Emergency: signal *Voice only, runner* / TIN
- **Control Unit:** the Incident Commander
- **Units:** all five Leadership members

Field Operations Net SOI

- **Objective:** Field team coordination and tasking
- **PACE:** OPS Primary / BRONZE · OPS Alternate / NICKEL · CMD Emergency / COPPER · Emergency: signal *Voice only, runner* / TIN
- **Control Unit:** the Operations Chief
- **Units:** all ten Field Operations members plus the Operations Chief

Each PACE tier is either a frequency or a plain signal. Clear the **Frequency** checkbox on a tier to describe it in words instead, which is how the Emergency tier above is set to a runner rather than a radio.

> An SOI carries its own SARNEG. Scepter generates it the first time the SOI is opened, and it is independent of the CEOI SARNEG list — which is why this CEOI has a SARNEG on each net even though the SARNEG count in Options is 0.

The **Assets** tab issues CEOI material to the net. Both SOIs here are issued the day-one DRYAD; because the count is 1 per division, both nets share the same table each day. The same can be done from the DRYAD list with the **Assignments...** menu item.

Step 9 — Print

Define a print job per group under **File** → **Settings** → **Print Jobs...** and run it with **File** → **Print...**. A Leadership packet holds the Command Net SOI and their call signs and authentication codes; a Field Operations packet holds the Field Operations Net SOI and theirs. Both should also carry:

- Call signs and authentication codes for every day of the period
- Both challenge/response sets for each day
- The DRYAD for each day
- The PACE plan from their SOI
- The frequency and location codes

Distribution and Security

- Print and hand out in person. Never email an unencrypted CEOI packet.
- Each operator receives only their own card.
- Cards are destroyed by tearing or burning when they expire, or immediately if capture is a risk.
- Keep one master copy secured for reference and re-issue.

Next Steps

- SARNEGs, for encoding numeric values in voice messages
- One-time pads, for encrypted text traffic
- Passwords, for equipment access and secondary authentication
- Document codes, so operational documents can be cited without naming them

Example 3 uses all of them.

Related Topics

- Getting Started Overview
- Example 1 — Simple
- Example 3 — Complex
- DRYADs
- Signals Operating Instructions
- Generation Modes
- DRYADs Report
- SOI Report

Example 3 — Complex: Full Operational CEOI

A preparedness organization of forty operators plans long-duration field operations using the whole communications security suite. It runs Scepter at its maximum configuration: 10 operational periods of 10 days each, giving 100 rotating code slots per issuance. The organization is built from five elements — Command, Alpha Team, Bravo Team, Logistics and Signals — operating three nets with full PACE plans, DRYADs and SARNEGs for field crypto, one-time pads for sensitive text traffic, and coded references for every frequency, location and document.

The finished CEOI ships with Scepter as `examples/example3.ceoi`. Every value below is the value in that file.

Scenario

Setting	Value	Why
Operators	40 across 5 groups	Full organizational structure
Periods	10 (maximum)	The whole issuance in one file
Period Divisions	10 (maximum)	100 code slots in total
Division Units	Day	Codes roll over daily
Call signs	1 per unit per day	Anonymous identification
Authentication codes	1 per unit per day	Check-in verification
Passwords	2 per unit per day (Command, Signals); 1 elsewhere	Equipment access and secondary authentication
Challenge / Response	3 sets per day	One per net
DRYADs	2 per day — 200 tables	High-volume grid coordinate traffic
SARNEGs	2 per day — 200 keywords	Voice encoding of numeric values
One-time pads	2 per day — 200 pads, DIANA, 250 characters	Encrypted text messages
SOIs	3	Command, Field Operations, Logistics
Frequencies	11, coded once per period	Three nets with backups
Locations	12, coded once per period	Every key site, with MGRS
Documents	6, coded once per period	Reference formats and standing orders

Step 1 — Cover and Options

On the **Cover** page:

- **Title:** OPERATIONS CEOI — Q3 2026
- **Organization:** Cascade Preparedness Association
- **Periods:** 10 · **Period Divisions:** 10 · **Division Units:** Day
- **Distribution:** Confidential, or whatever your classification policy calls for

Then work down **Options**:

Options page	Settings
Units	**Field A Label** = Role, **Field B Label** = Element
Frequencies	Mode **Period**, Pattern `LLN`, Count 1, Add to Word List on
Locations	Mode **Period**, Pattern `LLN`, Count 1, Add to Word List on, Report Detail Field **MGRS**
Documents	Mode **Period**, Pattern `LLN`, Count 1, Add to Word List on
DRYADS	Mode **Division**, Count **2**
SARNEGs	Mode **Division**, Count **2**
Challenge/Response	Mode **Division**, Count **3**
One Time Pads	Mode **Division**, Count **2**, Pad Type **DIANA**, Pad Size **250**

Step 2 — Frequencies

Name	Frequency	Mode	Purpose
CMD-P	147.180	FM	Command net primary
CMD-A	147.480	FM	Command net alternate
CMD-C	7.185	LSB	Command net contingency, HF NVIS
OPS-P	146.550	FM	Field operations primary
OPS-A	146.580	FM	Field operations alternate
OPS-C	446.075	FM	Field operations contingency, UHF
LOG-P	146.460	FM	Logistics primary
LOG-A	146.490	FM	Logistics alternate
REGIONAL	155.160	FM	County / regional emergency
ADMIN	446.000	FM	Administrative traffic

MEDEVAC	155.340	FM	Medical evacuation coordination

One code per frequency per period lets an operator say “move to Frequency TANGO” instead of announcing the frequency in the clear.

Step 3 — Locations

Twelve sites, each with MGRS and latitude/longitude:

- Primary Command Post, Alternate Command Post
- FOB North, FOB South
- Fuel Point, Resupply Point, Casualty Collection Point
- Observation Post 1, Checkpoint Alpha
- Rally Point 1, Extraction Point
- Regional Trauma Center

With location codes an operator can transmit “element is at LIMA SEVEN” instead of reading MGRS coordinates over an unencrypted voice channel.

Step 4 — Documents

Six references that operators may need to cite in field traffic. A document entry is a name and a description; attaching the actual file is optional.

- Standing Radio Procedures
- Medical Evacuation Request Format
- Contact Report Format
- Supply Request Format
- Communications Annex
- CEOI Amendment 1

One code per document per period lets an operator call for “Format SIERRA” without naming the document.

Step 5 — Units

Forty units, numbered sequentially by element: Command 1–6, Alpha 7–16, Bravo 17–26, Logistics 27–33, Signals 34–40. The two custom fields renamed in Options carry the rest:

- **Role** (Field A): Team Leader, Radio Operator, Medic, Crypto Custodian, and so on
- **Element** (Field B): Command, Alpha, Bravo, Logistics, Signals

Step 6 — Groups

Five groups issue codes, one per element. Each has **Generate Group Codes** off, **Generate Group Member Codes** on, and **Ensure Globally Unique** on.

Group	Callsign Pattern	Example	Authentication Code Pattern	Password Count
Command (6)	`LLN`	BK7	`NNNN`	2
Alpha Team (10 + 1 attached)	`LNN`	K47	`NNNN`	1
Bravo Team (10 + 1 attached)	`NLN`	4K7	`NNNN`	1
Logistics (7)	`NNNN`	4821	`NNNN`	1
Signals (7)	`LLNN`	BK47	`NNNN`	2

The five patterns differ in shape, not only in length, so a call sign heard on the air identifies the element it belongs to. Callsign Count and Authentication Code Count are 1 everywhere; Password Count is 2 for the two elements that hold crypto material and 1 for the rest.

> Every count on the Code Options tab is per unit *per division*. At 10 periods of 10 divisions, a Callsign Count of 1 is 100 call signs per unit, and a Password Count of 2 is 200 passwords per unit. This CEOI holds 4,200 call signs, 4,200 authentication codes and 5,500 passwords.

> Scepter refuses to generate unless the pattern has room for at least 1.5 times the codes requested. Command needs 6 × 100 = 600 call signs, so it needs 900 available codes; `LLN` holds 6,760 and fits, while a two-character `LN` holds only 260 and would be rejected. Widen the pattern whenever Scepter reports insufficient code space.

Multiple memberships and parent groups

Two Signals operators are attached out to the field teams, one to Alpha and one to Bravo. Each is a member of both Signals and their team, and so carries two call signs for every day — the correct behavior, since their identity depends on which net they are working.

A sixth group, **Field Element**, holds Alpha Team and Bravo Team as member groups using the **Member Of** tab. It issues no codes of its own: both Generate Group Codes and Generate Group Member Codes are off. It exists so material meant for both teams can be assigned and printed in one place.

Step 7 — Crypto Assets

DRYADs and SARNEGs

Two DRYAD tables and two SARNEG keywords per day, generated automatically: 200 of each across the ten periods. Brief operators on which table carries which traffic — for example DRYAD 1 for grid coordinates and DRYAD 2 for times, SARNEG 1 for unit strengths and SARNEG 2 for distances and bearings.

One-Time Pads

Pads are generated for the CEOI as a whole, not per operator: **Count** is the number of pads produced for each day, here 2, in DIANA format at 250 characters each. Individual pads are then issued to the operators who will carry them, using **Assignments...** on the One Time Pads page or the **Assets** tab of the unit editor. This example issues one pad to the Signals crypto custodian and one to the Command radio operator.

Direction is applied when the pad is printed, not when it is generated. The One Time Pad IN and One Time Pad OUT reports print the same pad under an inbound or outbound heading, so a station can keep its send and receive material apart.

> A one-time pad reused even once becomes vulnerable to cryptanalysis. Enforce strict pad discipline: each pad sheet is used once and destroyed immediately afterwards.

Step 8 — Signals Operating Instructions

Three SOIs, one per net. Each carries a comm schedule covering all ten days of the period.

Command Net

- **Objective:** Direction and reporting between Command and element leaders
- **PACE:** CMD-P / STEEL · CMD-A / IRON · CMD-C / COPPER · Emergency: *Voice only, runner* / TIN
- **Control Unit:** the Element Commander
- **Units:** all of Command plus the Signals net control operator
- **Comm Schedule:** 0600, 1200, 1800 and 2200, thirty minutes each

Field Operations Net

- **Objective:** Tasking and reporting for Alpha and Bravo
- **PACE:** OPS-P / GRANITE · OPS-A / BASALT · OPS-C / SLATE · Emergency: *Voice only, runner* / TIN
- **Control Unit:** the Signals net control operator
- **Units:** Alpha and Bravo plus the Command operations officer
- **Comm Schedule:** hourly from 0600 to 2200, fifteen minutes each

Logistics Net

- **Objective:** Resupply, fuel and casualty movement
- **PACE:** LOG-P / HARVEST · LOG-A / GRANARY · CMD-A / IRON · Emergency: *Voice only, runner* / TIN
- **Control Unit:** the Logistics chief
- **Units:** all of Logistics plus the Command supply officer
- **Comm Schedule:** 0700, 1300 and 1900, thirty minutes each

Each net is also issued its own material on the SOI **Assets** tab: the Command Net holds challenge/response set 1 and DRYAD 1 for the day, the Field Operations Net holds set 2 and DRYAD 2, and the Logistics Net holds set 3. Repeat the assignment for each day when the packets are prepared.

> The SARNEG shown on the SOI page belongs to that SOI. Scepter generates it when the SOI is first opened, and it is separate from the 200 SARNEGs in the CEOI SARNEG list, which are issued to units and groups as assets.

Step 9 — Print Operator Packets

With forty operators and 100 code slots, nobody carries the whole CEOI. Build a print job per element under **File** → **Settings** → **Print Jobs...** and run it with **File** → **Print...**:

- **Command packet:** call signs, authentication codes and passwords, the Command Net SOI, all DRYAD tables and SARNEGs, location, frequency and document codes, and their pads
- **Alpha / Bravo packet:** their call signs and authentication codes, the Field Operations Net SOI, their DRYADs and SARNEGs, and location and frequency codes
- **Logistics packet:** their call signs and authentication codes, the Logistics Net SOI, DRYAD tables, and location and frequency codes
- **Signals packet:** everything — Signals operators must be able to support any net

A print job is a saved configuration, so it can be re-run unchanged whenever the CEOI is regenerated or amended.

Accountability and Distribution

- Number each printed packet and record who received it in a distribution log.
- Require operators to sign for their packet. A lost packet is reported immediately and answered with an amendment: a new serial number and re-keyed crypto material.
- All material is destroyed at the end of the period or on compromise. Pad sheets are destroyed page by page as they are used.
- Keep the master Scepter file encrypted with a strong password (**File** → **Set Password ...**) and backed up offline somewhere secure.

Exercises and Validation

A complex CEOI is only useful if operators can work it under field conditions. Before it goes live:

- Run a tabletop exercise: everyone finds their call sign, builds a DRYAD-encoded message, and answers a challenge correctly.
- Run a radio exercise with the CEOI in effect — net check-ins, authentication, at least one DRYAD exchange, and one pad-encrypted message.
- Correct anything the exercises turn up before the CEOI is activated.

Related Topics

- Getting Started Overview
- Example 2 — Medium
- DRYADs
- SARNEGs

- One Time Pads
- Signals Operating Instructions
- Assigning Assets
- Generation Modes
- Generated Code Patterns
- Print Jobs

What is a CEOI?

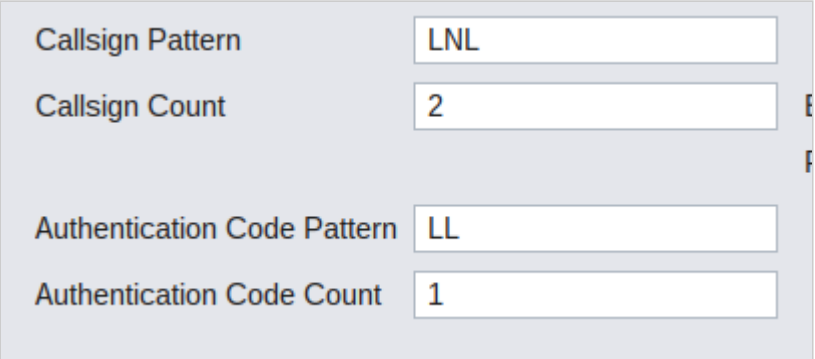
A **Communications-Electronics Operating Instructions** (CEOI) document — also historically called Signal Operating Instructions (SOI) — is a U.S. military combat order that provides the technical control and coordination of all communications within a command. It is the authoritative reference that every element in a unit needs in order to talk securely: who to call, on what frequency, what name to use, how to prove your identity, and what cipher to use when the message must be kept secret. A CEOI is issued for a defined operational period and is updated on a regular cycle (sometimes daily) so that compromised call signs, frequencies, or crypto material can be retired quickly. Each unit down the chain of command receives a version derived from the higher headquarters CEOI, ensuring that everyone operates with compatible — but appropriately compartmented — information.

In practice a CEOI bundles together several distinct types of material. **Call signs** identify each station on the radio net without revealing the unit's true designation. **Frequencies** tell operators where on the spectrum to transmit and receive for each net. **Authentication codes** and **challenge/response passwords** let stations verify that the person on the other end is who they claim to be and is not under duress. **DRYADs** provide a simple pencil-and-paper method for encrypting numerical data such as grid coordinates. **SARNEG (Search and Rescue Numerical Encryption Grid)** tables serve a similar purpose for encoding numeric values in search-and-rescue contexts. **One-time pads (OTPs)** offer provably unbreakable encryption for text messages: each pad is used exactly once, after which it is destroyed, leaving an adversary with no mathematical foothold for decryption. Scepter manages all of these elements together in a single CEOI file, generates the cryptographic material automatically according to your configuration, and produces print-ready output for distribution to the field.

Related Topics

- Main Window
- Generation Modes
- Generated Code Patterns
- Cover Settings
- DRYADs
- SARNEGs
- One Time Pads
- Challenge/Response

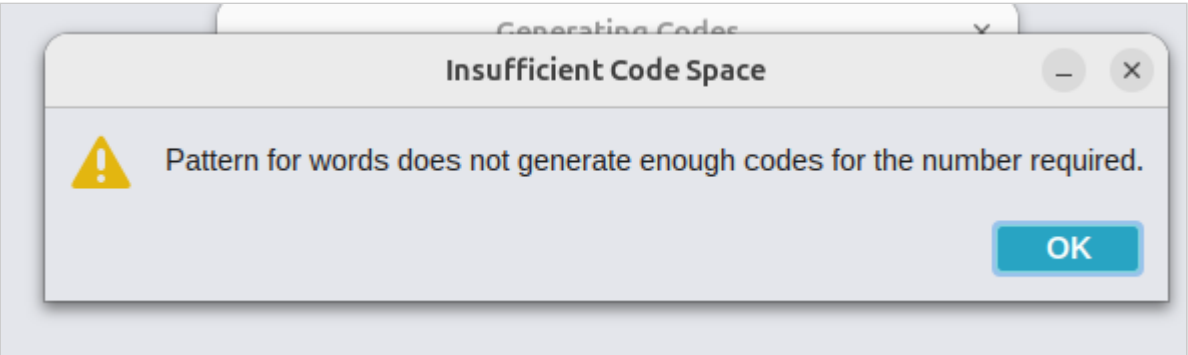
Generated Code Patterns



All the options that generate codes take a pattern to dictate how those codes look. There are 2 possible characters for those patterns

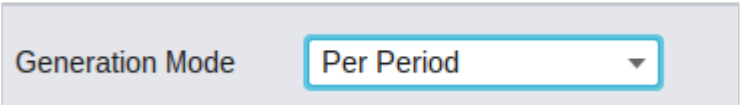
- 'L' For a letter A-Z
- 'N' For a number 0-9

Patterns are made up of repeated strings of 'L' and 'N'. So 'LNL' would be Letter Number Letter, e.g. K3T. Each pattern has a finite number of codes that can be created that match the pattern called the code space. If you specify a pattern that does not provide the required code space, for example: The pattern 'NNN' for a word list with over 1000 entries in it; you will get an error when attempting to generate codes.



Also keep in mind the count of codes to be generated as well as the generation mode as that will effect the available code space as well.

Generation Modes



Almost all the options that generate codes have a 'Generation Mode', the exception being units. Units are always 'Per Division'

- Per CEOI

  Generate 'count' codes for each item. (item count * size)

- Per Period

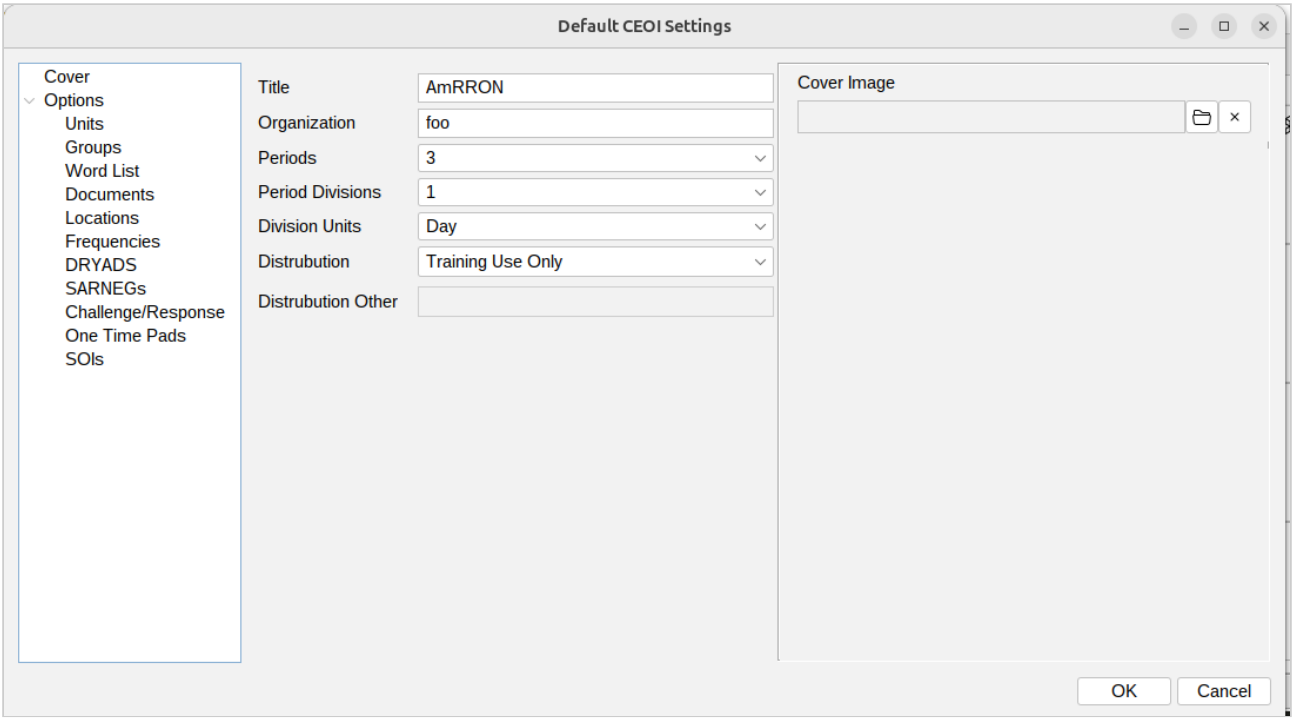
  Generate 'count' codes for each period in ceoi (#Periods * item count * size)

- Per Division

  Generate 'count' codes for each Division in ceoi (#Periods * #Divisions * item count * size)
  'Division' will display whatever the Division Units is set to on the Cover Settings

The generation mode will have an effect on the selection of patterns because they factor into the required code space values. See Generated Code Patterns

Default CEOI Settings



The Default CEOI Settings dialog lets you configure the default values that are applied when creating a new CEOI. Changes made here do not affect existing CEOIs — they only set the starting values for new ones.

Navigation

A tree menu on the left side of the dialog organizes settings into categories. Click a category to display its settings panel on the right:

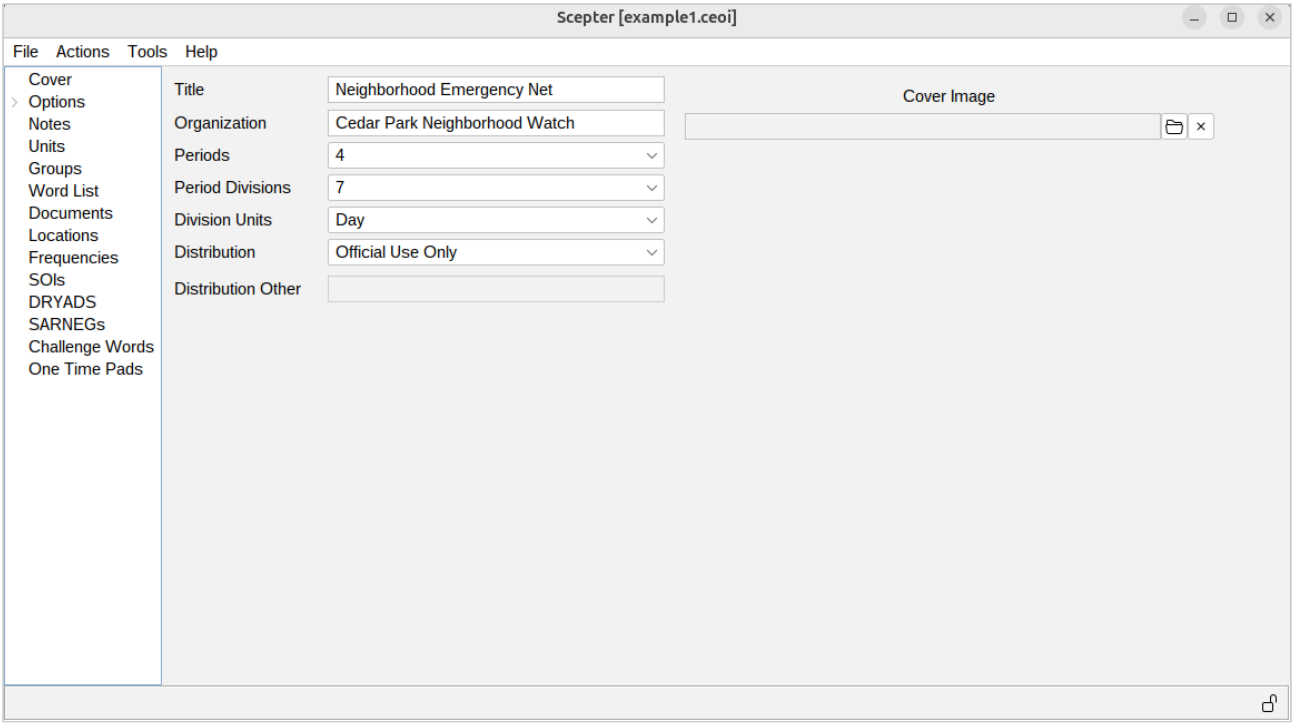
- **Cover** — Default cover page fields such as title, organization, period structure, and distribution level. See Cover Panel.
- **Options** — Default code generation and asset settings, with sub-categories for each entity type:
  - Units
  - Groups
  - Word List
  - Documents
  - Locations
  - Frequencies
  - DRYADs
  - SARNEGs
  - Challenge/Response
  - OTPs
  - SOI

Actions

Button	Description
OK	Saves all changes and closes the dialog. The new defaults will be used the next time a CEOI is created.
Cancel	Discards all changes and closes the dialog.

> These defaults are application-level settings. Each individual CEOI can still override any of these values through its own Options and Cover panels.

Cover Settings



The Cover Settings panel is the first panel in the navigation tree. It contains the basic identifying information for your CEOI, including its title, organization, time structure, classification, and an optional cover image.

Fields

Field	Description
Title	The name or title of the CEOI (e.g., "Operation Nightfall CEOI").
Organization	The unit or organization that owns the CEOI.
Periods	The number of time periods in the CEOI (1–10). Each period represents a distinct block of time during which a set of codes and assignments is valid.
Period Divisions	The number of divisions within each period (1–10). Divisions subdivide a period into smaller time blocks, each of which can have its own generated codes.
Division Units	The time unit for each division: Day, Week, Month, Quarter, or Year. This label appears throughout the application wherever division-level items are displayed.
Distribution	The classification or distribution level of the CEOI. Options include Training, Official Use Only, Confidential, Classified, Secret, Top Secret, or Other.
Distribution Other	A free-text field that becomes enabled when you select **Other** as the distribution level. Enter a custom distribution marking here.

Cover Image

The right side of the panel displays an optional cover image. This image appears on the printed CEOI cover page.

- Click the folder icon to browse for and load an image file (PNG, JPG, or BMP).
- Click the **X** button to remove the current cover image.
- Right-click on the image area and select **Save Cover Image to Disk** to export the current cover image to a file.

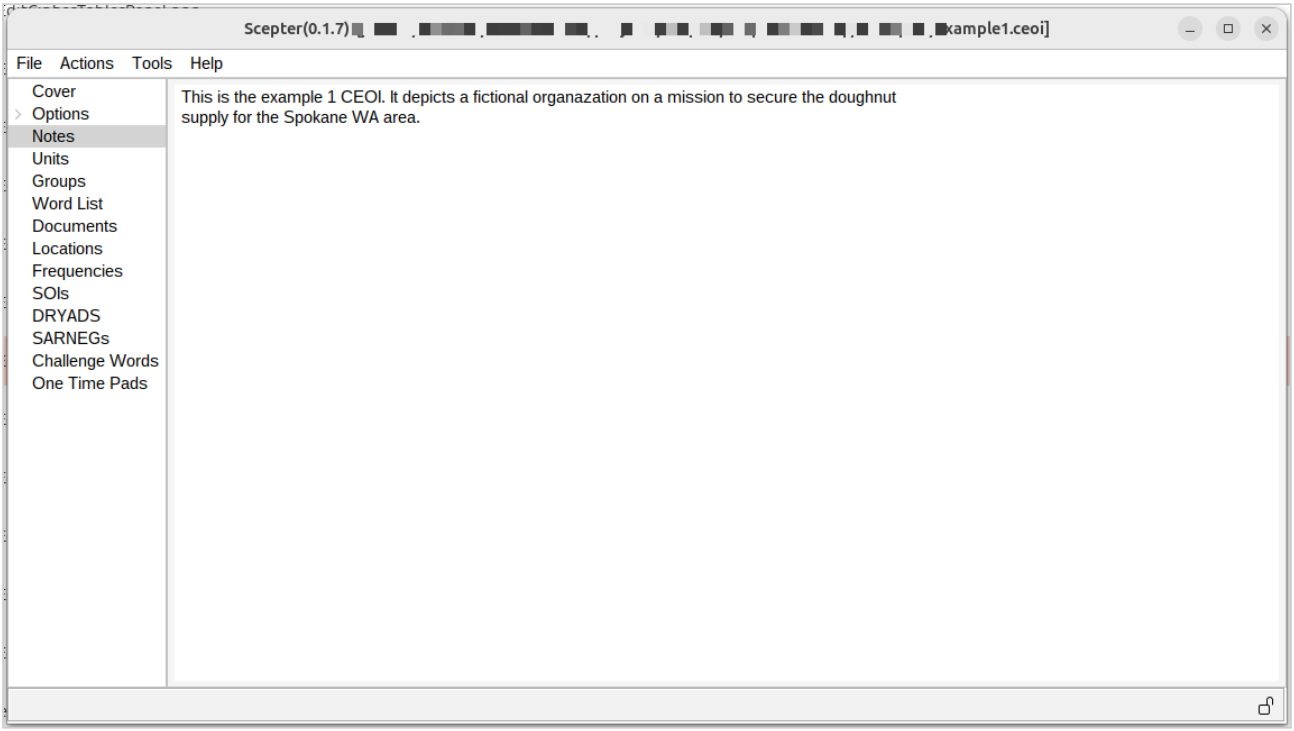
> The cover image is stored inside the CEOI file, so you do not need to keep the original image file on disk after loading it.

How Periods and Divisions Affect Code Generation

The number of periods, divisions, and the division unit setting directly affect how many codes are generated for each entity. The Generation Mode on each entity type's settings panel determines whether codes are generated per CEOI, per period, or per division.

> Changing the number of periods or divisions after generating crypto will require you to regenerate to reflect the new structure.

Notes



The Notes panel provides a free-text editor where you can add notes, comments, and supplementary information to the current CEOI. These notes are saved as part of the CEOI file and can be used to record planning details, standing instructions, distribution notes, or any other information relevant to the communications plan.

Using the Notes Editor

The panel contains a single text editing area that fills the entire panel. You can type or paste text directly into this area. The editor supports standard text editing operations:

- **Select All** – Press `Ctrl`+`A` to select all text.
- **Copy** – Press `Ctrl`+`C` to copy selected text to the clipboard.
- **Cut** – Press `Ctrl`+`X` to cut selected text.
- **Paste** – Press `Ctrl`+`V` to paste text from the clipboard.
- **Undo** – Press `Ctrl`+`Z` to undo the last edit.

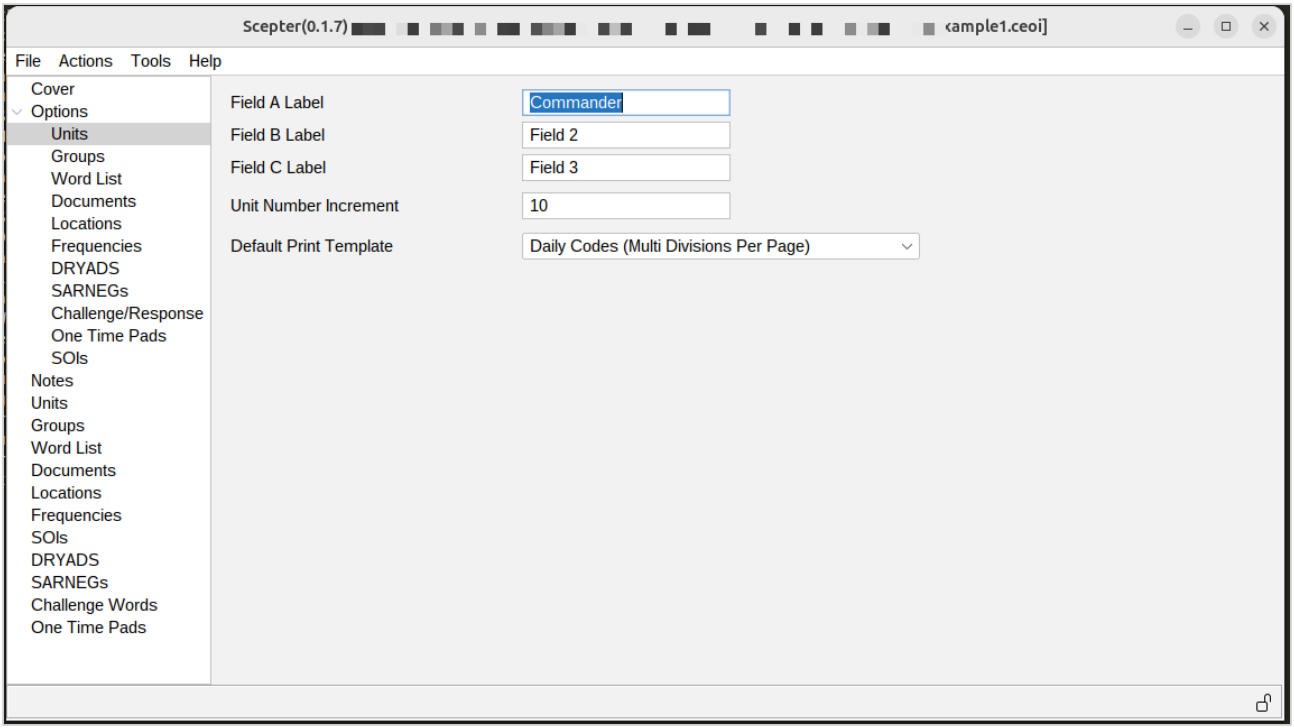
Saving Notes

Notes are saved automatically when you save the CEOI file. There is no separate save action for notes. Any text you enter or modify in the notes area is included the next time you save the CEOI via **File > Save** or `Ctrl`+`S`.

> Use the Notes panel to record distribution lists, special instructions, coordination details, or references to related orders and plans that do not fit into the structured CEOI fields.

The Notes panel is accessible from the **Notes** item in the navigation tree on the left side of the Main Window.

Options



The Options panel contains configuration settings that control how Scepter generates cryptographic material and codes for the current CEOI. Settings are organized into sub-panels, each dedicated to a specific asset type. You access the Options panel from the **Options** item in the navigation tree on the left side of the Main Window.

Navigating Settings Sub-Panels

Expanding the **Options** node in the navigation tree reveals the individual settings sub-panels. Selecting a sub-panel item switches the Options area to display that sub-panel's settings. The following sub-panels are available:

Sub-Panel	Description
Units	Configure custom field labels for unit records (Field A, Field B, Field C), the unit number increment value, and the default print template for units.
Groups	Set the default print template for groups.
Word List	Configure the generation mode, code type (sequence or random), code pattern, and code count for word list entries.
Documents	Configure the generation mode, code pattern, code count, and whether document codes are added to the word list.
Locations	Configure the generation mode, code pattern, code count, report detail field, coordinate preference, and whether location codes are added to the word list.
Frequencies	

	Configure the generation mode, code pattern, code count, and whether frequency codes are added to the word list.
DRYADs	Configure the generation mode, count of DRYADs to generate, and the default print template for DRYAD tables.
SARNEGs	Configure the generation mode, count of SARNEG tables to generate, and the default print template for SARNEGs.
Challenge/ Response	Configure the generation mode, count of challenge/response password pairs, and the default print template.
One Time Pads (OTPs)	Configure the generation mode, pad count, pad type, pad size, default print template, codex printing option, and numeric conversion tables (codexes) used with OTPs.
SOI	Configure the default SOI duration in days, per-day SARNEG/password/ DRYAD generation options, and the default print template for SOIs.

Common Settings

Many sub-panels share common settings fields:

Generation Mode

The Generation Mode determines how codes are assigned when you run **Actions > Generate Crypto**. See Generation Modes for a full explanation of each mode.

Pattern

The Pattern field defines the format of generated codes. Patterns use placeholder characters to describe the structure of the code. See Code Patterns for the pattern syntax.

Count

The Count field specifies how many codes to generate per entity. Valid values range from 0 to 100.

Add to Word List

When checked, codes generated for this asset type are also added to the CEOI word list, making them available for word list encoding and decoding.

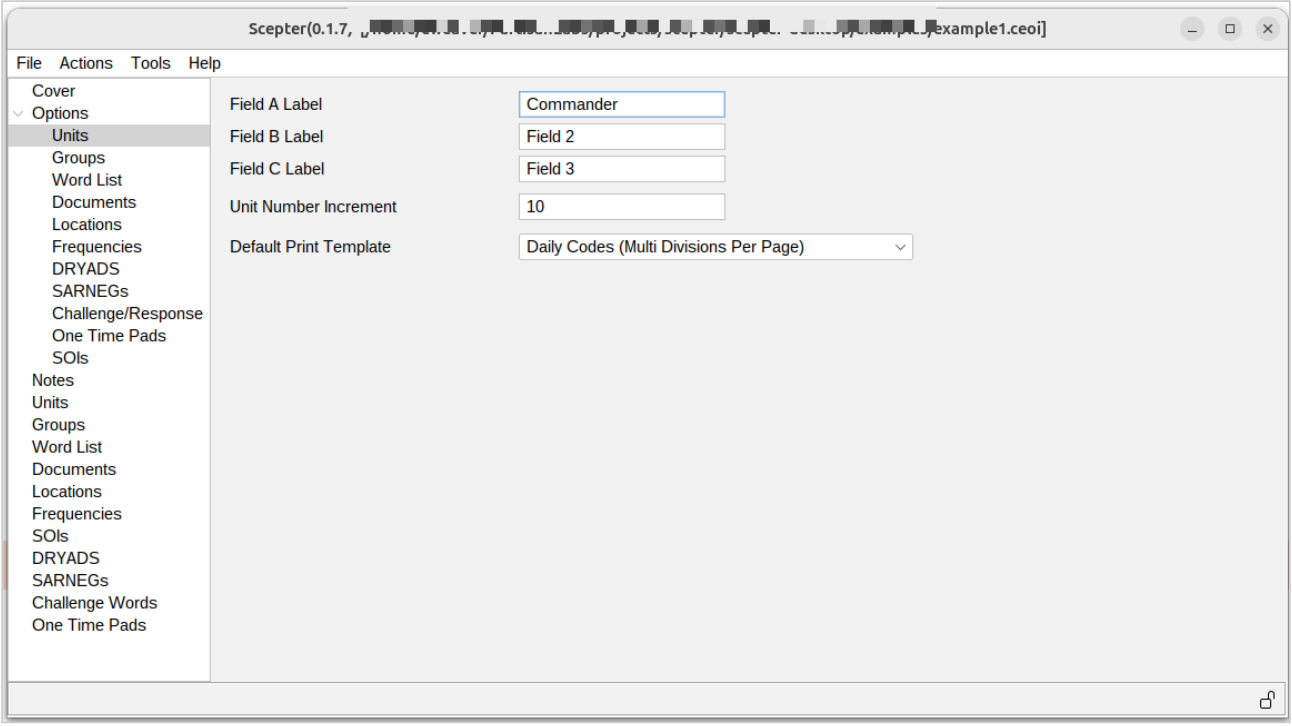
Default Print Template

Selects the print template used by default when printing this asset type. Print templates can be managed via **File > Settings > Print Templates**. See Print Templates.

> Changes to options take effect the next time you generate cryptographic material. If you have already generated crypto and want the new settings to apply, run **Actions > Remove Generated Crypto** first, then **Actions > Generate Crypto**.

> Default values for all options can be preset using **File > Settings > CEOI Default Settings**. See CEOI Defaults. Every new CEOI you create will start with those defaults.

Units Settings



The Units Settings panel configures how unit records are labeled and numbered in the current CEOI. Units are a core organizational element of the CEOI, representing individual stations or positions within the communications plan. This panel is accessed by selecting **Options > Units** in the navigation tree.

> Units always use the **Per Division** generation mode. There is no Generation Mode selector on this panel because unit codes (callsigns, authentication codes, and passwords) are always generated per division. See Generation Modes for more information.

Settings Reference

Field	Description
Field A Label	Sets the display label for the first custom field on unit records. This label appears as a column header in the units list and as a field label in the unit editor. Use it to identify the type of information stored in Field A (for example, "Role" or "Position").
Field B Label	Sets the display label for the second custom field on unit records, used the same way as Field A Label.
Field C Label	Sets the display label for the third custom field on unit records, used the same way as Field A Label.
Unit Number Increment	Controls the numeric step value used when auto-numbering units. For example, an increment of 10 assigns unit numbers 10, 20, 30, and so on. Valid values range from 1 to 100. The default value is 10.

Default Print Template	Selects the print template used by default when printing unit records. Print templates control the layout and formatting of printed output. See Print Templates for details on managing templates.

Custom Field Labels

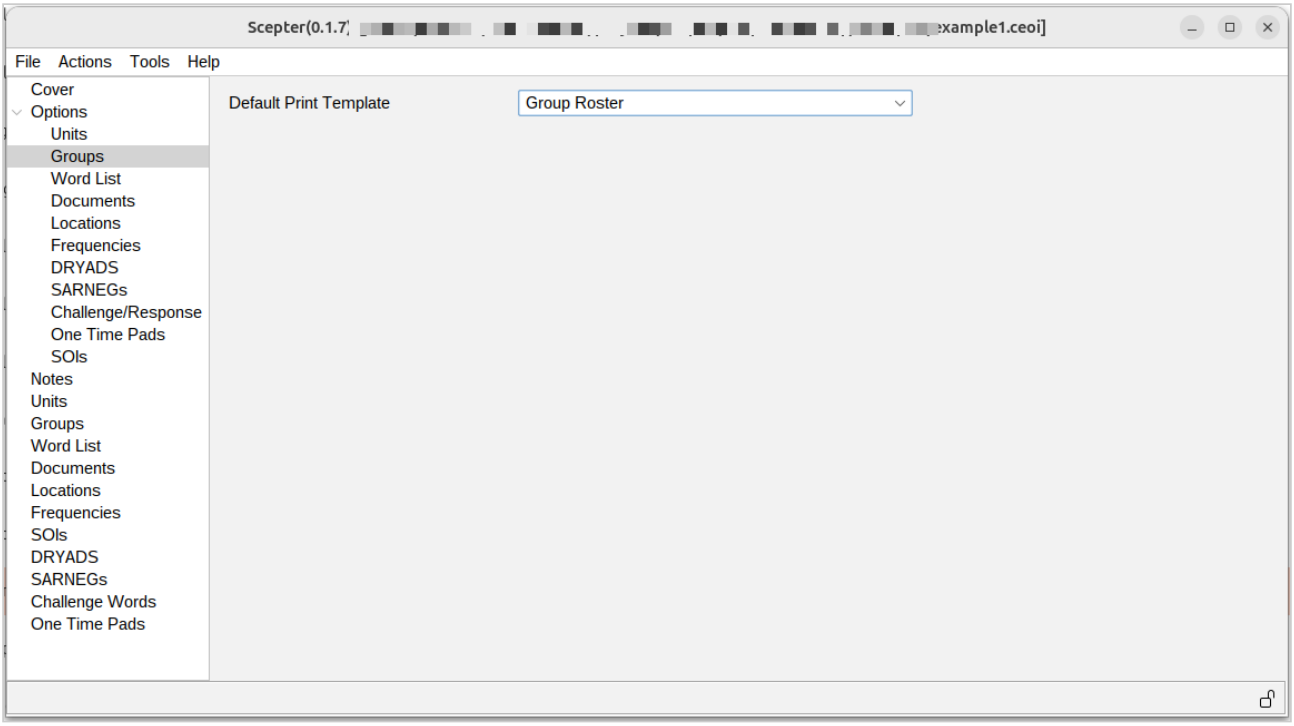
Each unit record in a CEOI has three user-defined fields (A, B, and C) that you can repurpose for any information relevant to your communications plan. The labels you set here are reflected throughout the application wherever unit records are displayed or edited, including the unit list, unit editor, and printed reports.

> If you do not need all three custom fields, you can leave unused field labels blank. The corresponding columns and fields will still appear but can be left empty on individual unit records.

Related Topics

- Options Overview
- Generation Modes
- Cover Settings -- where Division Units is configured
- Print Templates

Groups Settings



The Groups Settings panel configures the default print template for group records in the current CEOI. Groups represent collections of units that share a common communications net or organizational role. This panel is accessed by selecting **Options > Groups** in the navigation tree.

Settings Reference

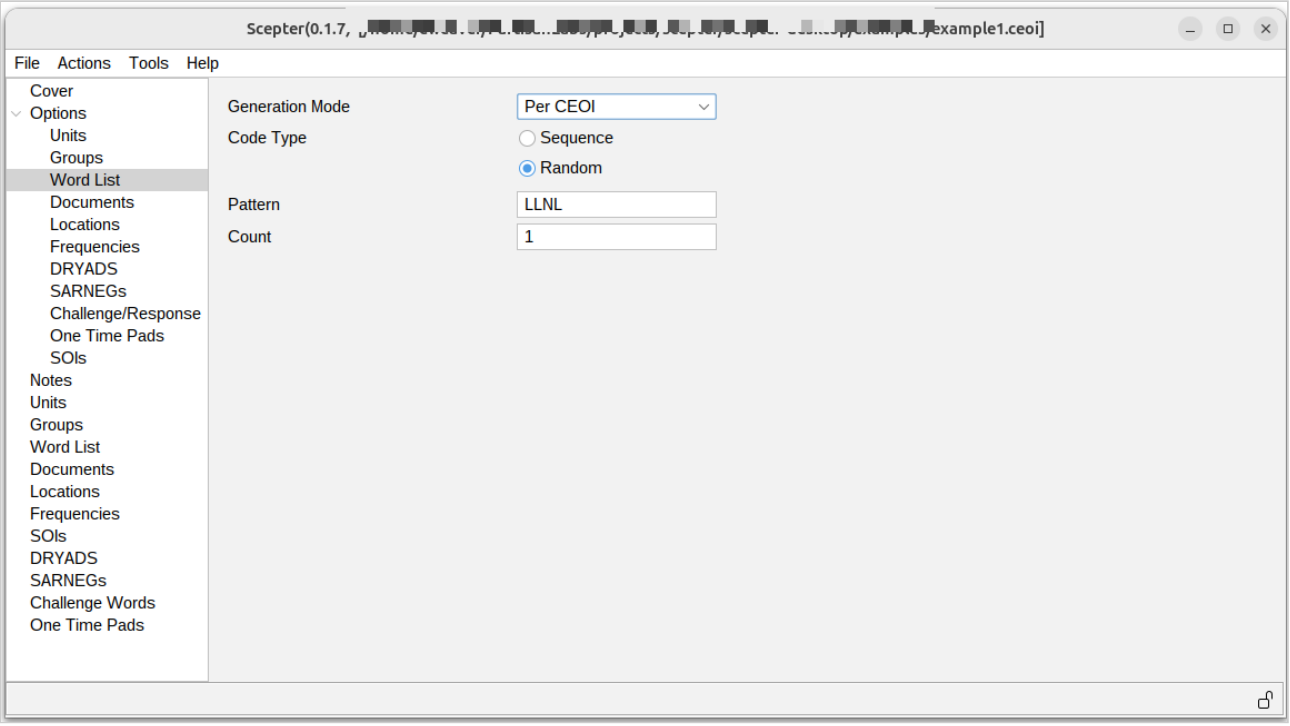
Field	Description
Default Print Template	Selects the print template used by default when printing group records. The template controls the layout and formatting of group information in printed output. See Print Templates for details on managing templates.

> Groups do not have their own code generation settings because group cryptographic material is derived from their member units. The callsigns, authentication codes, and other codes associated with a group come from the units assigned to it.

Related Topics

- Options Overview
- Units Settings
- Print Templates

Word List Settings



The Word List Settings panel configures how codes are generated for word list entries in the current CEOI. The word list is a vocabulary of words and phrases used for encoding and decoding short messages. Each word in the list is assigned a code during crypto generation. This panel is accessed by selecting **Options > Word List** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how word list codes are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Code Type	Determines how codes are assigned to words: • **Sequence** -- Assigns codes in sequential order. When this option is selected, the Pattern and Count fields are disabled because the codes follow a fixed numeric sequence rather than a generated pattern. • **Random** -- Generates random codes matching the specified pattern. This is the default. The Pattern and Count fields are enabled to control the generated code format.
Pattern	Defines the format of generated codes using placeholder characters. Only enabled when Code Type is set to Random. See Code Patterns for the pattern syntax.
Count	

	Specifies how many codes to generate per word list entry. Only enabled when Code Type is set to Random. Valid values range from 0 to 100. The default is 1.

Code Type Behavior

When you select **Sequence** as the Code Type, each word in the list receives a sequential numeric code (for example, 001, 002, 003). This is a compact encoding scheme but the codes are predictable.

When you select **Random** as the Code Type, each word receives one or more randomly generated codes matching the pattern you specify. Random codes are less predictable, improving communications security. Make sure the pattern provides enough code space for the word list size multiplied by the count. See Code Patterns for details on code space.

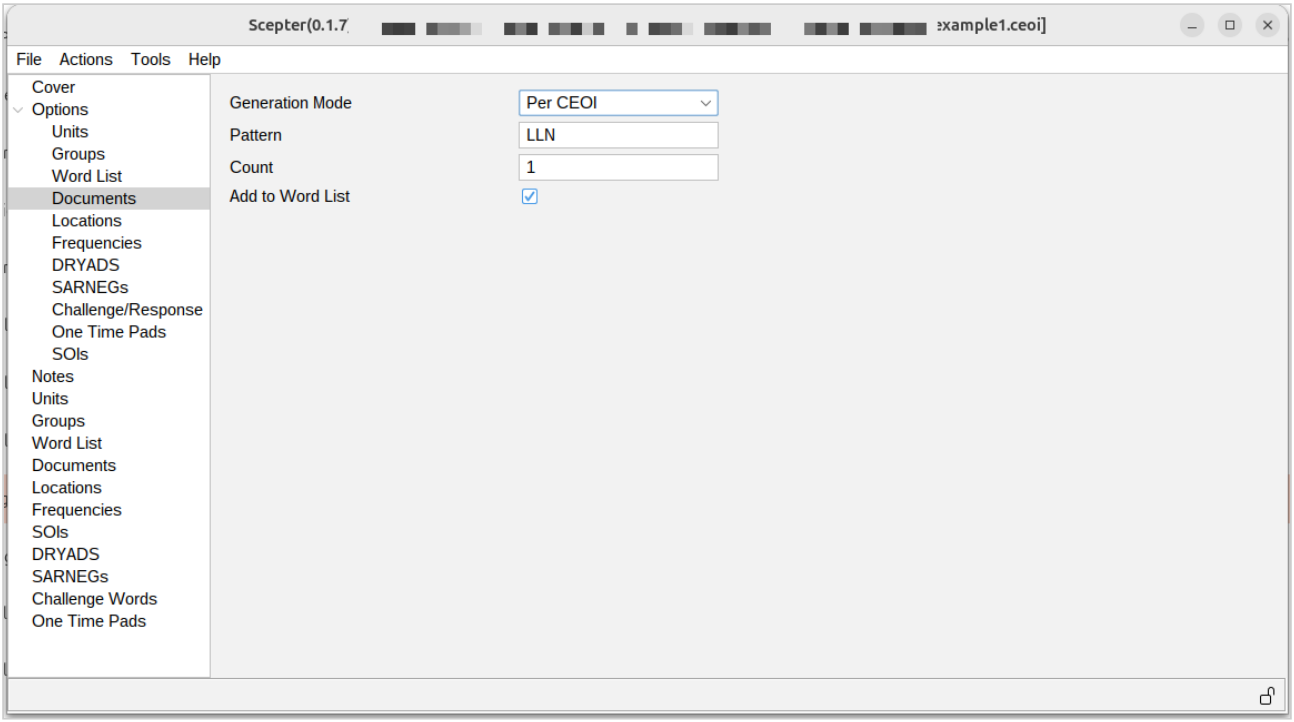
> If the pattern does not provide enough unique codes for the total number required (word count multiplied by code count, factored by the generation mode), you will receive an "insufficient code space" error when generating crypto. Increase the pattern length or reduce the count to resolve this.

> Other asset types such as Documents, Locations, and Frequencies can optionally add their codes to the word list using the "Add to Word List" checkbox on their respective settings panels. This allows those items to be referenced by word list codes as well.

Related Topics

- Options Overview
- Generation Modes
- Code Patterns
- Documents Settings
- Locations Settings
- Frequencies Settings

Documents Settings



The Documents Settings panel configures how codes are generated for document entries in the current CEOI. Documents represent files, publications, maps, or other reference materials that need to be identified by code in communications. This panel is accessed by selecting **Options > Documents** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how document codes are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Pattern	Defines the format of generated codes using placeholder characters. See Code Patterns for the pattern syntax.
Count	Specifies how many codes to generate per document entry. Valid values range from 0 to 100. The default is 1.
Add to Word List	When checked, codes generated for documents are also added to the CEOI word list. This makes document codes available for word list encoding and decoding, allowing documents to be referenced by their word list code. Enabled by default.

Add to Word List

Enabling **Add to Word List** means that each document and its generated code(s) will appear in the word list alongside regular word entries. This is useful when you want documents to be part of the

general encoding vocabulary, allowing operators to reference documents using the same code lookup used for words.

If you prefer to keep document codes separate from the word list, uncheck this option. Document codes will still be generated and available in the documents section of the CEOI, but they will not appear in the word list.

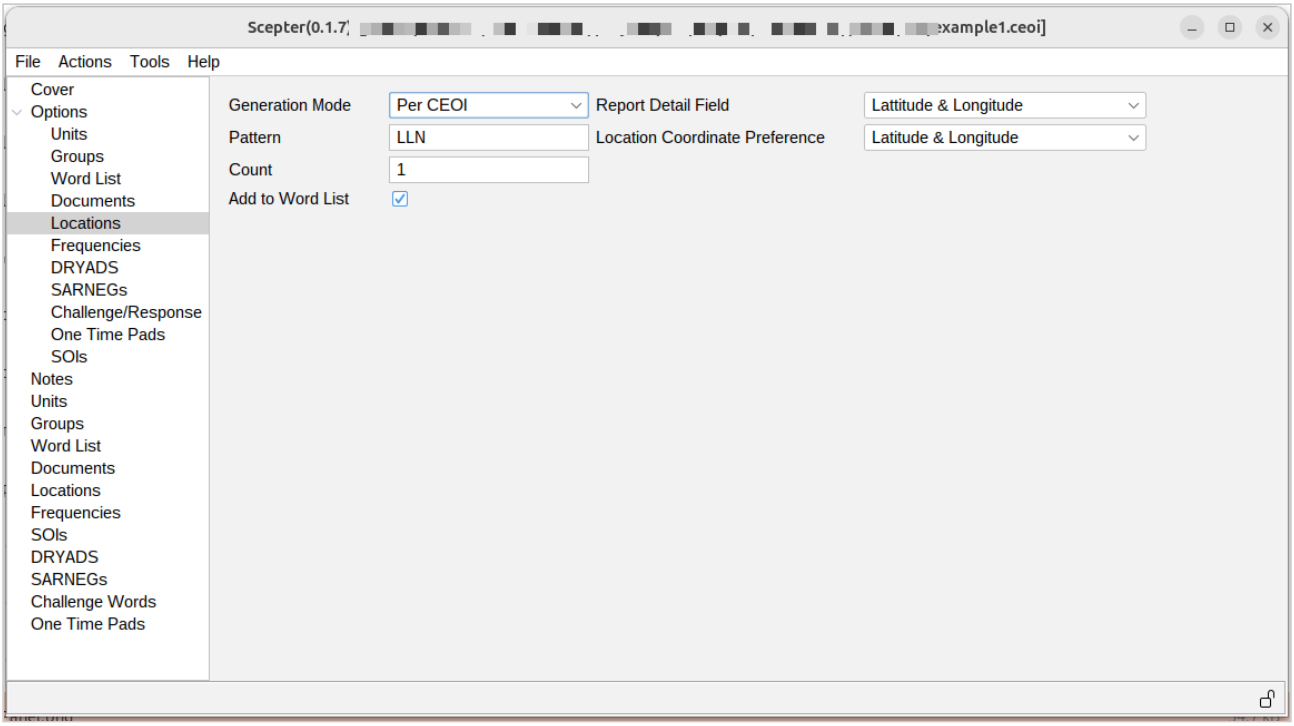
> If the pattern does not provide enough unique codes for the total number required (document count multiplied by code count, factored by the generation mode), you will receive an "insufficient code space" error when generating crypto. Increase the pattern length or reduce the count to resolve this.

> When Add to Word List is enabled, the document codes share code space with the word list. Make sure your word list pattern provides enough room for the combined total of word list entries plus any documents, locations, and frequencies that are also added to the word list.

Related Topics

- Options Overview
- Generation Modes
- Code Patterns
- Word List Settings

Locations Settings



The Locations Settings panel configures how codes are generated for location entries in the current CEOI and how location coordinate data is displayed and reported. Locations represent geographic positions such as checkpoints, rally points, or named areas of interest that need to be identified by code in communications. This panel is accessed by selecting **Options > Locations** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how location codes are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Pattern	Defines the format of generated codes using placeholder characters. See Code Patterns for the pattern syntax.
Count	Specifies how many codes to generate per location entry. Valid values range from 0 to 100. The default is 1.
Add to Word List	When checked, codes generated for locations are also added to the CEOI word list. This makes location codes available for word list encoding and decoding, allowing locations to be referenced by their word list code. Enabled by default.
Report Detail Field	Selects which coordinate format is shown in the detail column of location reports and printed output. See the Report Detail Field section below for available formats.

Location Coordinate Preference	Sets the preferred coordinate system used for entering and displaying location coordinates throughout the application. See the Coordinate Preference section below for available options.

Report Detail Field

The Report Detail Field controls which representation of a location's coordinates appears in the detail column of reports and printed CEOI output. Each location can store coordinates in multiple formats; this setting determines which one is displayed. The available options are:

Option	Description
Latitude & Longitude	Standard geographic coordinates in decimal degrees.
MGRS	Military Grid Reference System coordinates, commonly used in military operations.
UTM	Universal Transverse Mercator coordinates.
Maidenhead	Maidenhead Locator System grid squares, commonly used in amateur radio.
What 3 Words	What3Words geocoding system, which assigns a three-word address to every location.
BIP 39 Mnemonic (lat/lon)	BIP 39 mnemonic word encoding of latitude and longitude coordinates.

Location Coordinate Preference

The Location Coordinate Preference sets the preferred coordinate system for entering and viewing location data throughout the application. This affects how coordinate fields are presented in the location editor and other views. The available options are:

Option	Description
Latitude & Longitude	Standard geographic coordinates.
Military Grid Reference System (MGRS)	NATO standard grid reference system.
Universal Transverse Mercator (UTM)	UTM zone, easting, and northing coordinates.
Maidenhead Grid System	Maidenhead Locator System grid squares.
BIP 39 Mnemonic (lat/lon)	BIP 39 mnemonic word encoding.
No Preference	No specific coordinate system is preferred; all formats are treated equally.

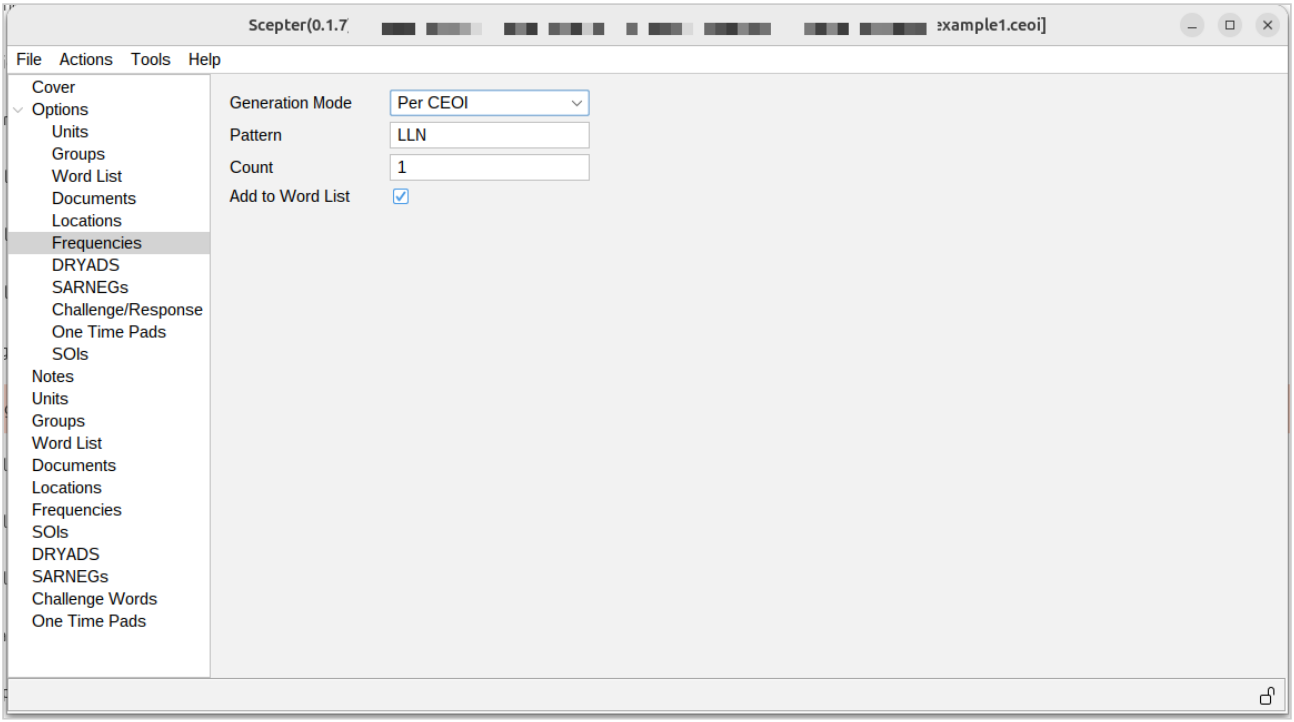
The Report Detail Field and Location Coordinate Preference are separate settings. The Report Detail Field controls what appears in printed reports, while the Coordinate Preference affects data entry and display within the application. You can set them to different formats if needed.

If the pattern does not provide enough unique codes for the total number required (location count multiplied by code count, factored by the generation mode), you will receive an "insufficient code space" error when generating crypto. Increase the pattern length or reduce the count to resolve this.

Related Topics

- Options Overview
- Generation Modes
- Code Patterns
- Word List Settings

Frequencies Settings



The Frequencies Settings panel configures how codes are generated for frequency entries in the current CEOI. Frequencies represent radio channels and their associated parameters that need to be identified by code in communications. This panel is accessed by selecting **Options > Frequencies** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how frequency codes are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Pattern	Defines the format of generated codes using placeholder characters. See Code Patterns for the pattern syntax.
Count	Specifies how many codes to generate per frequency entry. Valid values range from 0 to 100. The default is 1.
Add to Word List	When checked, codes generated for frequencies are also added to the CEOI word list. This makes frequency codes available for word list encoding and decoding, allowing frequencies to be referenced by their word list code. Enabled by default.

Add to Word List

Enabling **Add to Word List** means that each frequency and its generated code(s) will appear in the word list alongside regular word entries. This allows operators to encode references to specific radio frequencies using the same code lookup process used for ordinary words.

If you prefer to keep frequency codes separate from the word list, uncheck this option. Frequency codes will still be generated and available in the frequencies section of the CEOI, but they will not appear in the word list.

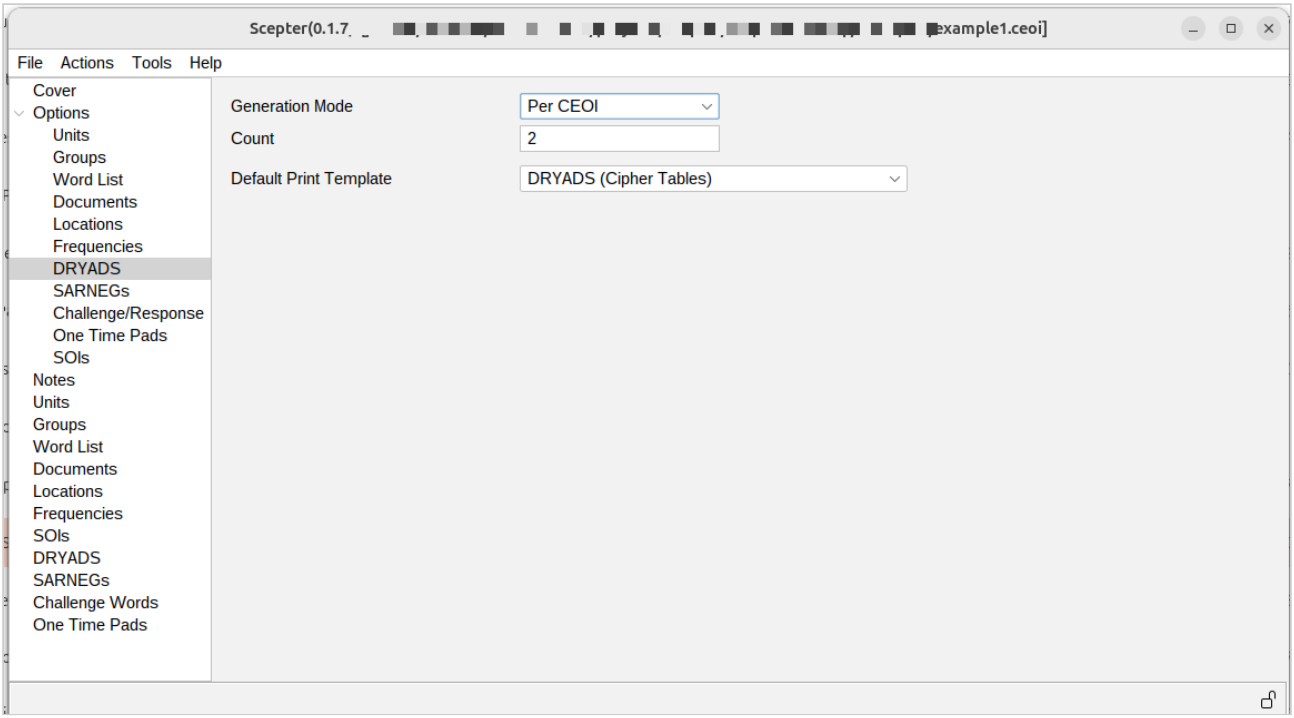
> If the pattern does not provide enough unique codes for the total number required (frequency count multiplied by code count, factored by the generation mode), you will receive an "insufficient code space" error when generating crypto. Increase the pattern length or reduce the count to resolve this.

> When Add to Word List is enabled, the frequency codes share code space with the word list. Make sure your word list pattern provides enough room for the combined total of word list entries plus any documents, locations, and frequencies that are also added to the word list.

Related Topics

- Options Overview
- Generation Modes
- Code Patterns
- Word List Settings

DRYAD Settings



The DRYAD Settings panel configures how DRYADs are generated for the current CEOI. DRYAD tables are numerical DRYADs used to encrypt grid coordinates and short numeric messages. Each table maps letter groups (ABC, DEF, GHI, etc.) to digit pairs, providing a quick field-encryption tool for tactical communications. This panel is accessed by selecting **Options > DRYADs** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how DRYADs are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Count	Specifies how many DRYAD tables to generate per generation mode unit. Valid values range from 0 to 100. The default is 1. For example, if Generation Mode is set to Per Period and Count is 2, two distinct DRYAD tables will be created for each period in the CEOI.
Default Print Template	Selects the print template used when printing DRYAD tables. The available templates are configured in the application's print settings. The selected template controls the layout and formatting of printed DRYAD table output.

How DRYAD Tables Work

A DRYAD has 25 rows, labeled A to Y. Each row holds the letters A to Y once each, in a scrambled order, split into ten columns headed by the digits 0 to 9. Columns 0, 1, 2 and 5 hold more letters than the others, so the commonest digits have more letters to choose from:

```
  ABCD EFG HIJ KL MN OPQ RS TU VW XY
  0    1   2   3  4  5   6  7  8  9
A NTFW XQH YJG PA SI OVL DR CB UM EK
```

The digit heading is what encryption uses. To encrypt a number, the sender picks the row for the message and, for each digit, sends one of the letters in that digit's column, choosing a different one when the digit repeats. The receiver reads each letter's column heading to get the digits back. The letter heading above the digits is A to Y split the same way, and turns a letter, such as a grid letter in a map reference, into the digit of the column it heads.

DRYADs are also used to authenticate a station: the challenger sends a row letter and a letter from that row, and the reply is the letter directly below it. DRYAD tables give a fast, low-technology way to encrypt grid coordinates, checkpoint numbers and other short numbers in the field.

Generation Mode and Count

The combination of Generation Mode and Count determines the total number of DRYAD tables produced. When the mode is set to Per Division, a separate set of tables is generated for each division in each period, providing the highest level of compartmentalization. Per Period creates one set per period shared across divisions, and Per CEOI creates a single set for the entire CEOI.

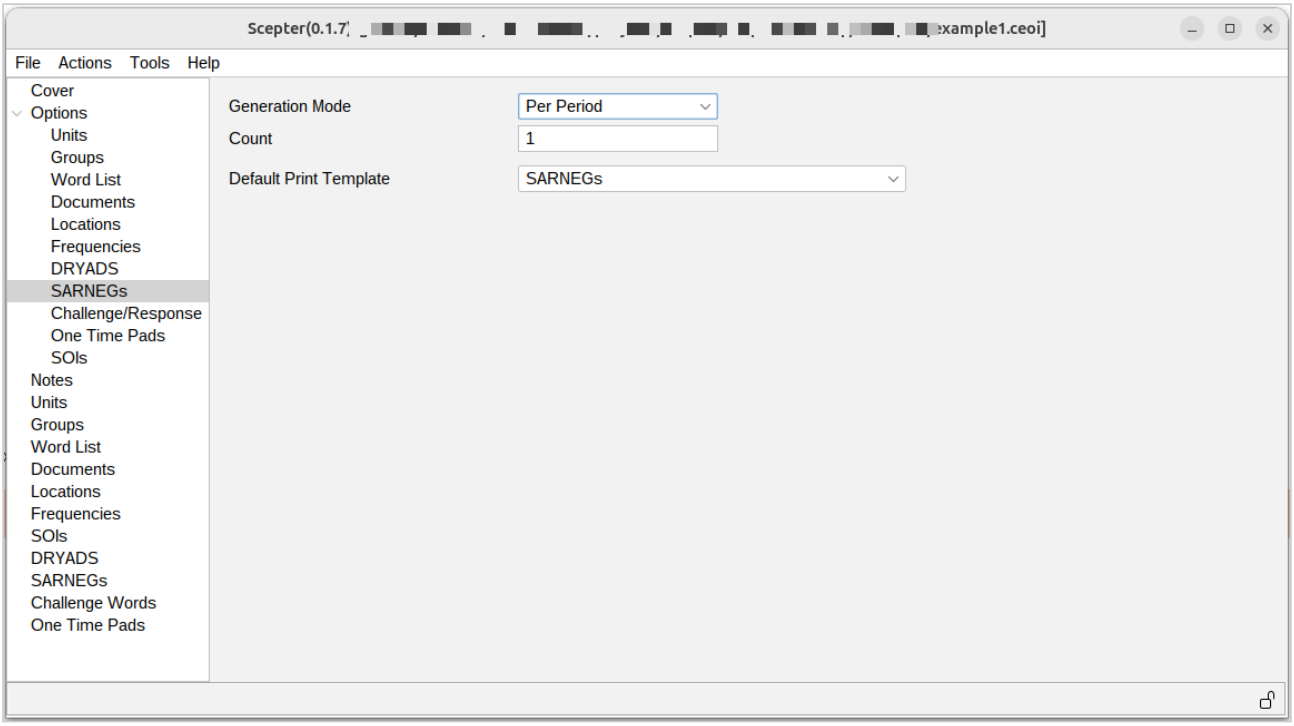
> DRYAD tables do not use code patterns because they generate their own internal numeric cipher mappings rather than alphanumeric identification codes. The Count field is the primary way to control the volume of generated tables.

> DRYAD tables can also be assigned on a per-day basis within SOI settings. See SOI Settings for the "DRYAD per day" option.

Related Topics

- Options Overview
- Generation Modes
- SOI Settings
- SARNEG Settings
- One-Time Pad Settings

SARNEG Settings



The SARNEG Settings panel configures how SARNEGs (Search and Rescue Numeric Encryption Groups) are generated for the current CEOI. A SARNEG is a code word used for authentication and encryption in tactical communications. Each SARNEG is a word whose individual letters can be used as a simple substitution cipher for digits. This panel is accessed by selecting **Options > SARNEGs** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how SARNEGs are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Count	Specifies how many SARNEGs to generate per generation mode unit. Valid values range from 0 to 100. The default is 1. For example, if Generation Mode is set to Per Period and Count is 2, two SARNEGs will be generated for each period in the CEOI.
Default Print Template	Selects the print template used when printing SARNEGs. The available templates are configured in the application's print settings. The selected template controls the layout and formatting of printed SARNEG output.

How SARNEGs Work

A SARNEG is a code word where each letter position corresponds to a digit (0-9). To encrypt a number, you substitute each digit with the letter at that position in the SARNEG word. For example, if the SARNEG is "BLACKHORSE" then the letter B = 0, L = 1, A = 2, and so on. The grid coordinate "327" would be encrypted as "CAR".

SARNEGs provide a fast, low-technology method for encrypting short numeric data such as grid coordinates, checkpoint numbers, and frequencies over voice communications. They are typically changed on a regular schedule to maintain security.

Generation Mode and Count

The Generation Mode and Count together determine how many SARNEGs are produced and how they are organized. Using Per Division mode provides each division with its own SARNEG for each period, giving the greatest compartmentalization. Per Period shares the same SARNEG across all divisions in a given period, and Per CEOI uses a single set of SARNEGs for the entire CEOI.

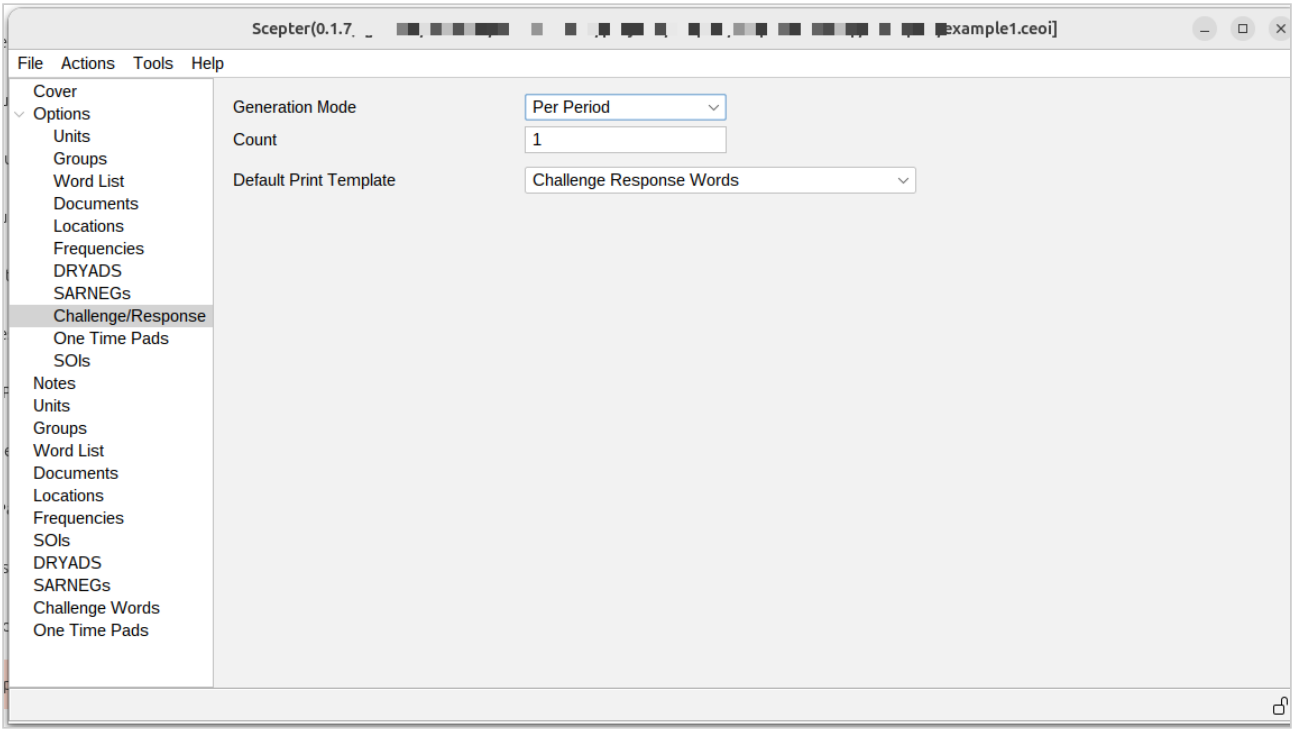
> SARNEGs are generated from a word source and do not use code patterns. The system automatically selects words of the appropriate length for SARNEG use. Each SARNEG word must have enough unique characters to map all digits 0 through 9.

> SARNEGs can also be assigned on a per-day basis within SOI settings. See SOI Settings for the "SARNEG per day" option.

Related Topics

- Options Overview
- Generation Modes
- SOI Settings
- DRYAD Settings
- Challenge/Response Settings

Challenge/Response Settings



The Challenge/Response Settings panel configures how challenge and response authentication pairs are generated for the current CEOI. Challenge/response pairs are used to verify the identity of personnel at checkpoints, over radio, or during link-up operations. Each pair consists of a challenge word, a correct response word, and a duress word. This panel is accessed by selecting **Options > Challenge/Response** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how challenge/response pairs are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Count	Specifies how many challenge/response pairs to generate per generation mode unit. Valid values range from 0 to 100. The default is 1. For example, if Generation Mode is set to Per Period and Count is 3, three separate challenge/response pairs will be generated for each period in the CEOI.
Default Print Template	Selects the print template used when printing challenge/response pairs. The available templates are configured in the application's print settings. The selected template controls the layout and formatting of printed challenge/response output.

How Challenge/Response Pairs Work

A challenge/response pair is a set of three words used for identity authentication:

- **Challenge Word** -- The word spoken by the person initiating the authentication. This is typically given at a checkpoint, over radio, or during a meeting.
- **Response Word** -- The correct reply that confirms the identity of the person being challenged. Only personnel with access to the current CEOI will know the correct response.
- **Duress Word** -- An alternative response word that signals the person is responding under coercion or duress. If the duress word is given instead of the normal response, it alerts the challenger that the situation is compromised while appearing normal to the captor.

Generation Mode and Count

The Generation Mode and Count together determine the total number of challenge/response pairs generated and how they are organized within the CEOI. Using Per Division mode provides each division with its own unique pairs for each period, offering the greatest isolation. Per Period shares the same pairs across all divisions within a given period, and Per CEOI uses a single set of pairs for the entire CEOI.

Setting a higher Count gives operators multiple pairs to choose from during each generation mode unit, which can be useful for rotating pairs throughout a duty day or assigning different pairs to different checkpoints.

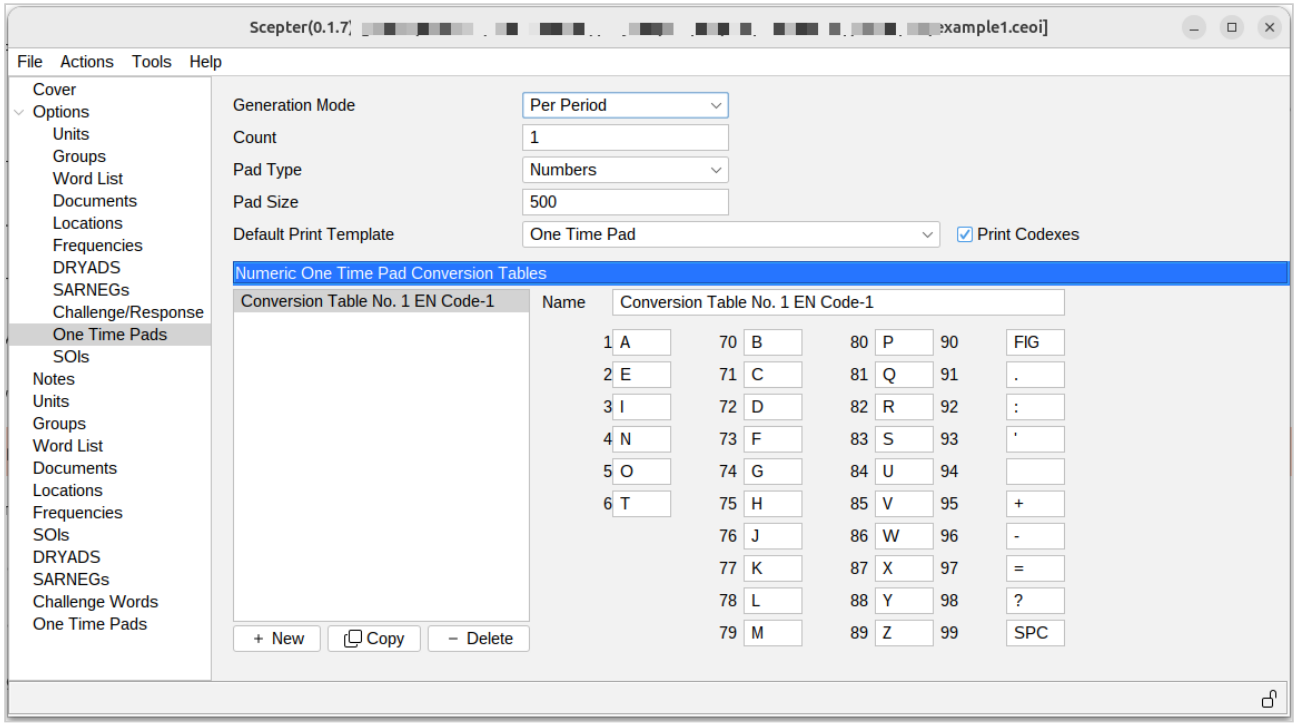
> Challenge/response pairs are generated from a word source and do not use code patterns. The system automatically selects words that are suitable for verbal authentication -- words that are easy to speak and hear clearly over radio or in field conditions.

> Challenge/response pairs can also be assigned on a per-day basis within SOI settings. See SOI Settings for the "Password per day" option, which controls whether a new challenge/response set is assigned for each day of the SOI duration.

Related Topics

- Options Overview
- Generation Modes
- SOI Settings
- SARNEG Settings
- DRYAD Settings

One-Time Pad Settings



The One-Time Pad (OTP) Settings panel configures how one-time pads are generated for the current CEOI. One-time pads provide theoretically unbreakable encryption when used correctly -- each pad is a sequence of random characters used to encrypt a single message and then destroyed. This panel is accessed by selecting **Options > OTPs** in the navigation tree.

Settings Reference

Field	Description
Generation Mode	Controls how one-time pads are distributed across the CEOI structure. Options are Per CEOI, Per Period, and Per Division. See Generation Modes for a detailed explanation of each mode.
Count	Specifies how many one-time pads to generate per generation mode unit. Valid values range from 0 to 100. The default is 1. For example, if Generation Mode is set to Per Division and Count is 3, three pads will be generated for each division in each period.
Pad Type	Determines the character set used for generating pad content. See the Pad Types section below for details on each option.
Pad Size	Sets the number of characters in each generated pad. The default is 512. Larger pads can encrypt longer messages but require more material to distribute and manage. The minimum pad size is 10 characters.
Default Print Template	Selects the print template used when printing one-time pads. The available templates are configured in the application's print settings.

Print Codexes	When checked, codex conversion tables are included alongside the one-time pad when printing. Codexes are needed for numeric pad types to convert between letters and numbers during encryption and decryption. See the Codex Conversion Tables section below.

Pad Types

The Pad Type setting determines what characters the generated pad contains. Each type is suited to different encryption methods:

Pad Type	Characters	Description
Numbers	0-9	Generates pads containing only numeric digits. Numeric pads require a codex conversion table to encrypt alphabetic messages -- you first convert each letter to its numeric equivalent using the codex, then add the pad digit modulo 10.
Letters (DIANA)	A-Z	Generates pads containing only uppercase letters. DIANA is a letter-based one-time pad system that uses a trigraph table (a DIANA conversion table) to combine plaintext letters with pad letters directly, without a separate numeric conversion step.
Numbers and Letters (DIANA+)	A-Z, 0-9	Generates pads containing a mix of both letters and digits. This extended DIANA format supports encrypting messages that contain both text and numbers without requiring a separate codex for numeric content.

Codex Conversion Tables

The lower portion of the OTP settings panel contains the **Numeric One Time Pad Conversion Tables** editor. A codex is a mapping table that assigns a two-digit number to each letter of the alphabet, plus special characters. These tables are used with numeric one-time pads to convert plaintext letters to numbers before encryption, and to convert decrypted numbers back to letters.

The codex editor displays a grid of number-to-letter mappings organized by ranges (1-6, 70-79, 80-89, 90-99). Each entry in the grid maps a specific number to a letter or special character.

Managing Codexes

- **New** -- Creates a new blank codex and adds it to the list. You can then enter a name and fill in the letter mappings.
- **Copy** -- Duplicates the currently selected codex.
- **Delete** -- Removes the currently selected codex from the list.

Select a codex from the list on the left to load its mappings into the editing grid on the right. The **Name** field at the top of the grid sets the display name for the codex. Edit individual cells in the grid to assign letters or special characters to each numeric position.

Codex conversion tables are only required for numeric pad types. If you are using DIANA or DIANA+ pads, the encryption is performed using a trigraph table instead of a codex, and the codex section is not used during encryption.

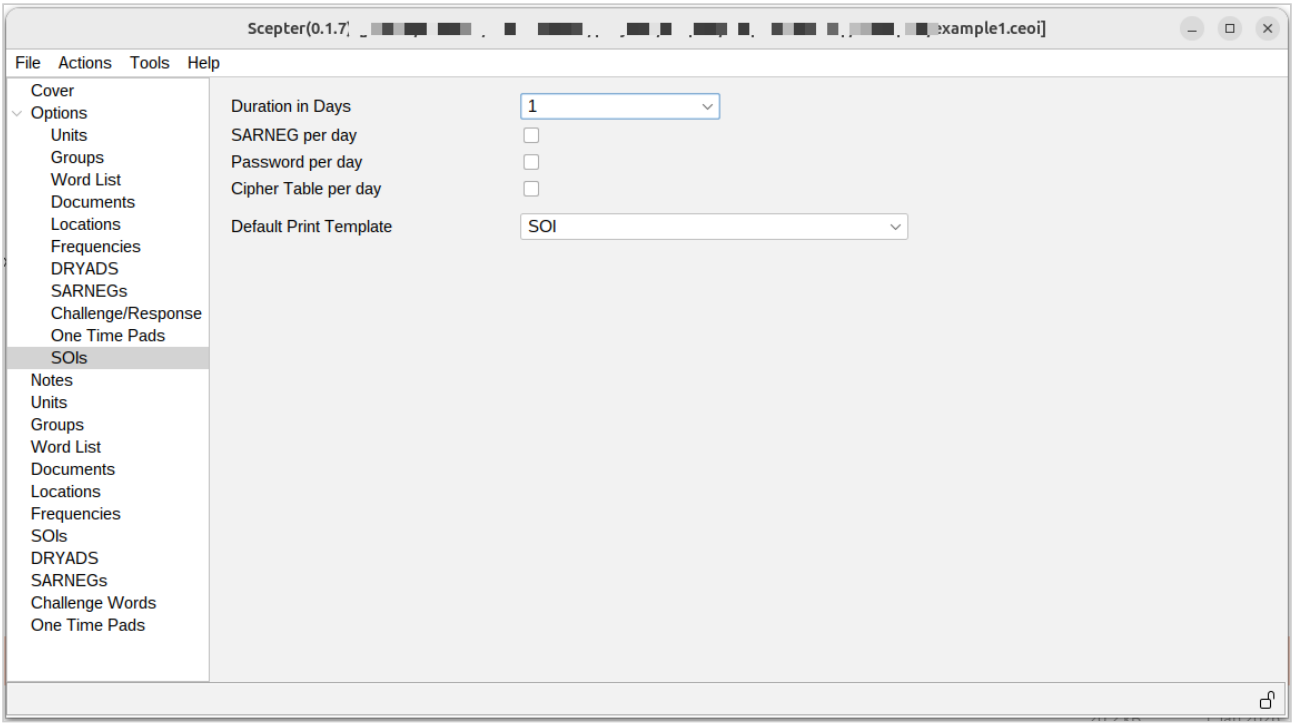
Enable the **Print Codexes** checkbox to include the codex conversion tables on printed OTP output. This ensures that operators in the field have the codex reference they need to use numeric pads.

One-time pads must never be reused. Each section of a pad should be used for a single message and then marked as consumed. The pad tracks used portions internally -- once characters are consumed they are replaced with asterisks (*) and cannot be used again.

Related Topics

- Options Overview
- Generation Modes
- DRYAD Settings
- SARNEG Settings
- Challenge/Response Settings

SOI Settings



The SOI (Signal Operating Instructions) Settings panel configures generation options for SOI documents in the current CEOI. An SOI is a comprehensive communications plan that brings together frequencies, call signs, authentication materials, and scheduling information into a single operational document. This panel is accessed by selecting **Options > SOI** in the navigation tree.

Settings Reference

Field	Description
Duration in Days	Sets how many days the SOI covers. Valid values are 1 through 10. This determines the number of day-specific entries generated for items configured to change daily (such as SARNEGs, passwords, and DRYADs). For example, a duration of 5 means the SOI will contain up to 5 separate daily sets of authentication materials.
SARNEG per day	When checked, a different SARNEG is assigned for each day of the SOI duration. When unchecked, the same SARNEG is used for the entire SOI period. Enabling this improves security by rotating the numeric encryption group daily.
Password per day	When checked, a different challenge/response pair (password) is assigned for each day of the SOI duration. When unchecked, the same challenge/response pair is used for the entire SOI period. Enabling this improves authentication security by rotating passwords daily.
DRYAD per day	

	When checked, a different DRYAD is assigned for each day of the SOI duration. When unchecked, the same DRYAD is used for the entire SOI period. Enabling this improves encryption security by rotating DRYADs daily.
Default Print Template	Selects the print template used when printing SOI documents. The available templates are configured in the application's print settings. The selected template controls the layout and formatting of printed SOI output.

SOI Duration

The Duration in Days setting is the foundation of the SOI configuration. It defines the operational time window the SOI covers. A longer duration means fewer SOI changes are required but provides more material that could be compromised. A shorter duration improves security but requires more frequent SOI distribution.

The duration directly affects items marked for daily rotation. For example, if the duration is 7 days and "SARNEG per day" is enabled, the SOI will include 7 different SARNEGs, one for each day. If "SARNEG per day" is not enabled, a single SARNEG covers all 7 days.

Daily Rotation Options

The three daily rotation checkboxes control whether specific cryptographic materials change on each day of the SOI period:

- **SARNEG per day** -- Rotates the numeric encryption group used for encrypting grid coordinates and short numeric data. See SARNEG Settings.
- **Password per day** -- Rotates the challenge/response authentication pairs used at checkpoints and during link-up operations. See Challenge/Response Settings.
- **DRYAD per day** -- Rotates the DRYAD used for numeric encryption. See DRYAD Settings.

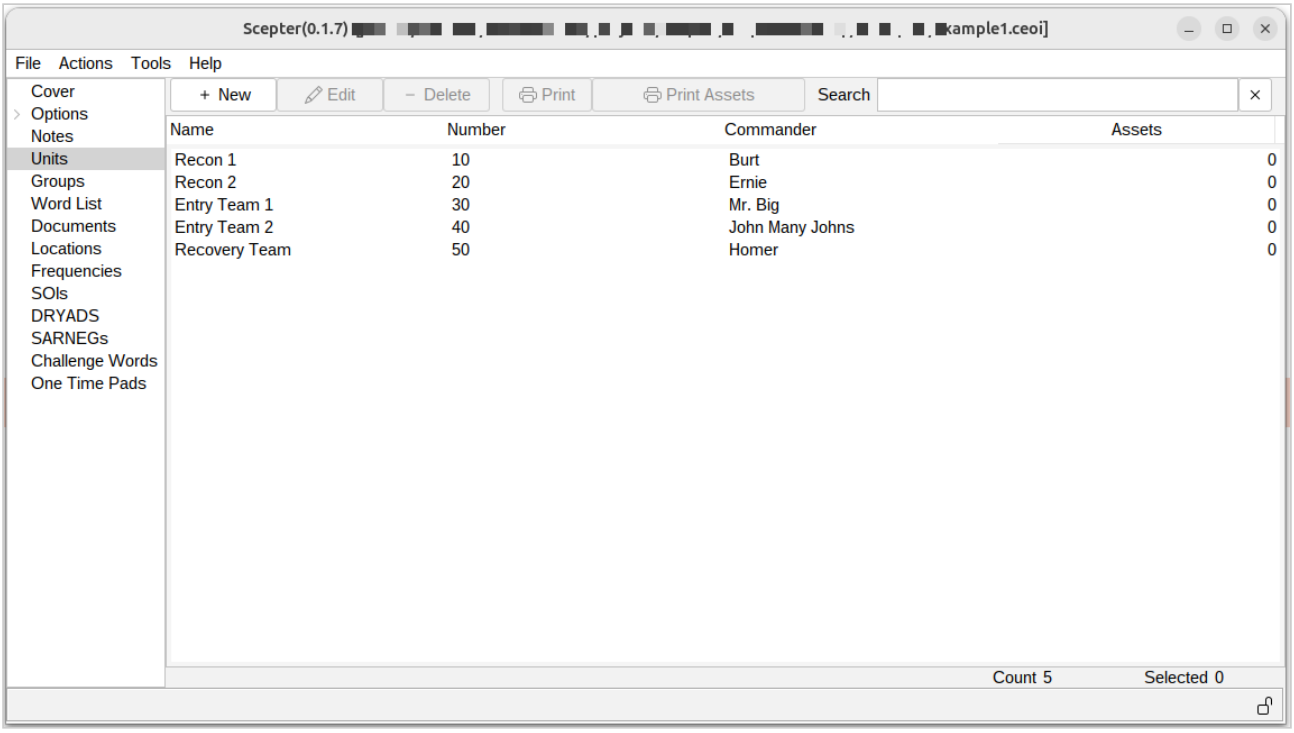
> Enabling daily rotation for all three authentication materials provides the best security posture. Even if one day's materials are compromised, the remaining days of the SOI remain secure.

> The SOI brings together materials generated by other settings panels. Make sure you have configured the SARNEG, Challenge/Response, and DRYAD settings with appropriate generation modes and counts before creating SOI documents. The SOI will draw from the pool of generated materials based on those settings.

Related Topics

- Options Overview
- SARNEG Settings
- Challenge/Response Settings
- DRYAD Settings
- One-Time Pad Settings

Units



The Units panel displays all units defined in the current CEOI. Units represent the individual teams, stations, or elements that participate in the communications plan. Each unit can be assigned callsigns, authentication codes, passwords, and other cryptographic material.

Table Columns

Column	Description
Name	The name of the unit.
Number	The unit's assigned number.
Field A	A custom field whose label is configured in the Cover Settings.
Assets	The number of assets (frequencies, locations, etc.) assigned to this unit.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new unit. The Unit Edit Dialog opens for you to enter the unit's details.
- **Edit** – Select a unit and click the pencil button, press `Enter`, or double-click the row to open the Unit Edit Dialog.
- **Delete** – Select one or more units and click the **−** button or press `Delete`. Deleting a unit also removes it from any groups it belongs to.
- **Print Assets** – Select one or more units and click the printer button to print asset assignments for the selected units.

Search

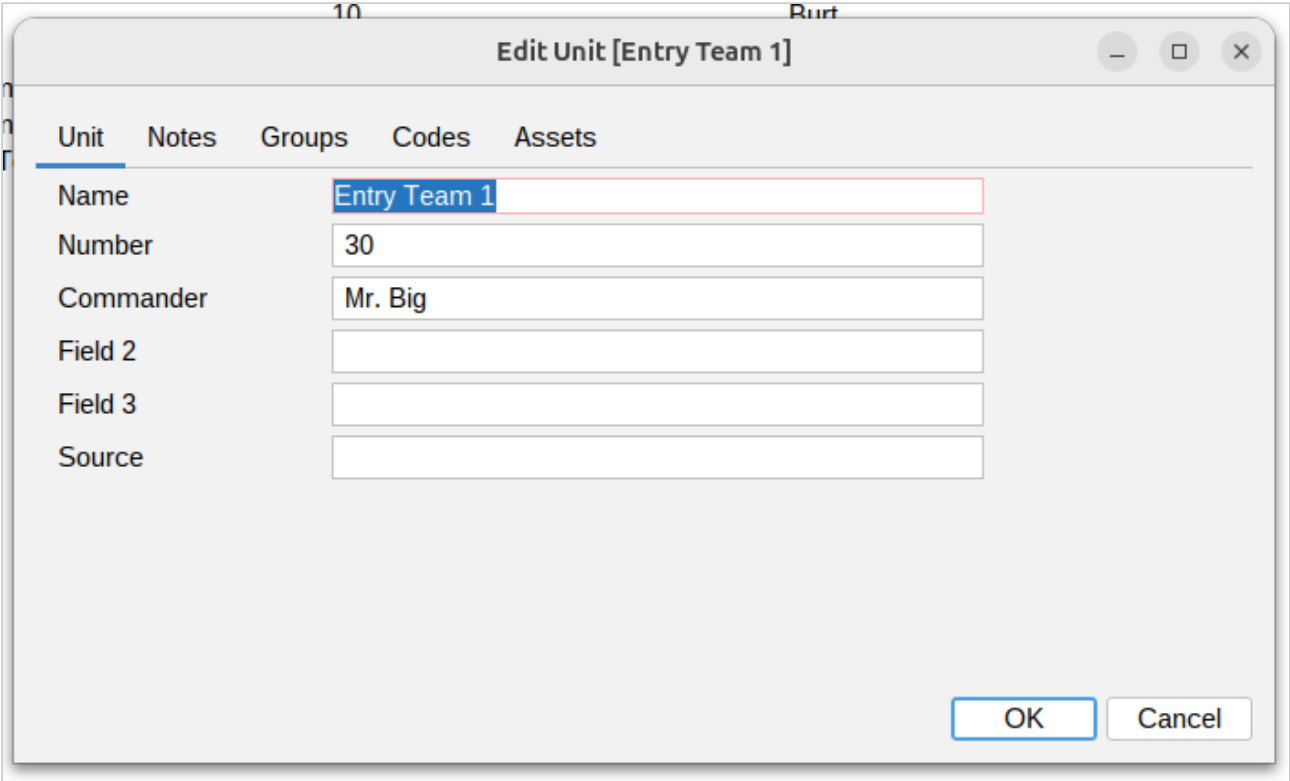
Type in the search field to filter the list. Units whose name starts with the entered text are shown; all others are hidden. Clear the search field to show all units again.

Import and Export

You can import a list of units from a file or export the current unit list using the **Tools > Import List** and **Tools > Export List** menu items (or right-click context menu) when this panel is active.

> The status bar at the bottom of the panel shows the total number of units and how many are currently selected.

Unit Edit Dialog



The Unit Edit Dialog is used to create and edit units in the current CEOI. Units represent individual teams, stations, or elements that participate in the communications plan. You open this dialog by adding a new unit or editing an existing unit from the Units panel.

When you create a new unit, the dialog title shows "New Unit" and the unit is automatically added to the All Units group. When you edit an existing unit, the dialog title shows "Edit Unit" followed by the unit's name.

Tabs

The dialog is organized into a set of tabbed panels, each providing access to a different aspect of the unit:

Tab	Description
Unit	Basic unit properties including name, number, custom fields, and source.
Notes	Free-text notes associated with the unit.
Groups	Group memberships for the unit. Every unit belongs to the All Units group by default.
Codes	Generated codes for the unit, organized into Callsigns, Auth Codes, Passwords, and Overrides sub-tabs. Codes are shown per group membership.
Assets	Assets (frequencies, locations, etc.) assigned to this unit.

Dialog Buttons

- **OK** -- Validates the unit data, saves changes, and closes the dialog. The unit name is required; if it is missing, a validation error is shown and the Unit tab is selected automatically.
- **Cancel** -- Discards all changes and closes the dialog. You can also press `Escape` to cancel.
- **Clear ALL Codes** -- Visible only when the Codes tab is selected. Removes all callsigns, authentication codes, and passwords for the currently selected group membership after a confirmation prompt.

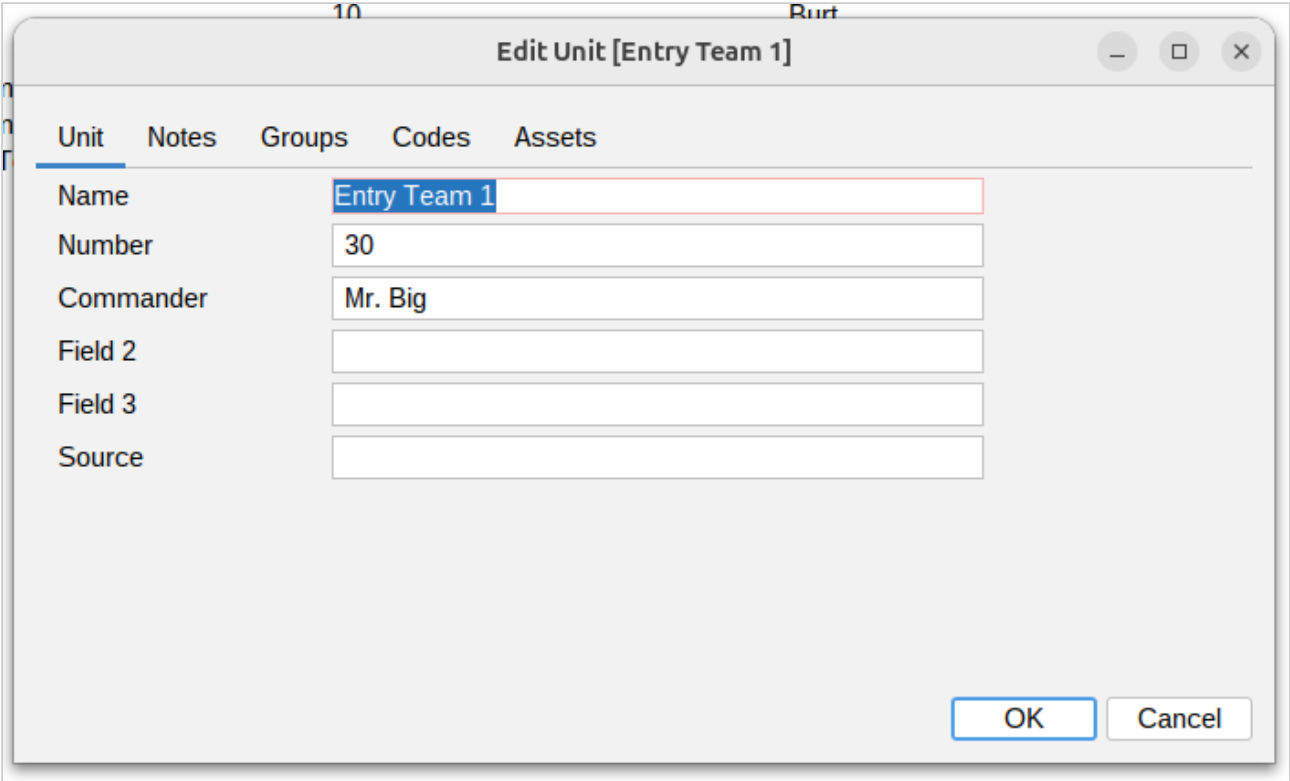
> Changes are not saved to the CEOI until you click **OK**. If you cancel or close the dialog, no changes are applied.

> You can press `Enter` to confirm the dialog at any time -- the OK button is the default button.

Related Topics

- Units Panel
- Units Settings
- Unit Tab
- Notes Tab
- Groups Tab
- Codes Tab

Unit Tab



The Unit tab is the first tab in the Unit Edit Dialog. It displays the basic properties of the unit, including its name, number, custom fields, and source.

Fields Reference

Field	Description
Name	The name of the unit. This field is required -- you cannot save the unit without providing a name. The name field is highlighted with a pink border to indicate it is mandatory.
Number	A numeric identifier for the unit. If you do not enter a value, the next available unit number is assigned automatically. Valid values range from 0 to 1,000,000. The auto-numbering increment is configured in Units Settings.
Field A	A user-defined custom field. The label shown for this field is configured in Units Settings (for example, "Role" or "Position"). If no label has been set, the default label "Field A" is displayed.
Field B	A second user-defined custom field. The label shown for this field is configured in Units Settings
Field C	A third user-defined custom field. The label shown for this field is configured in Units Settings
Source	Records the origin or source of this unit record. This is typically set automatically when units are imported from a file, but you can enter or edit it manually.

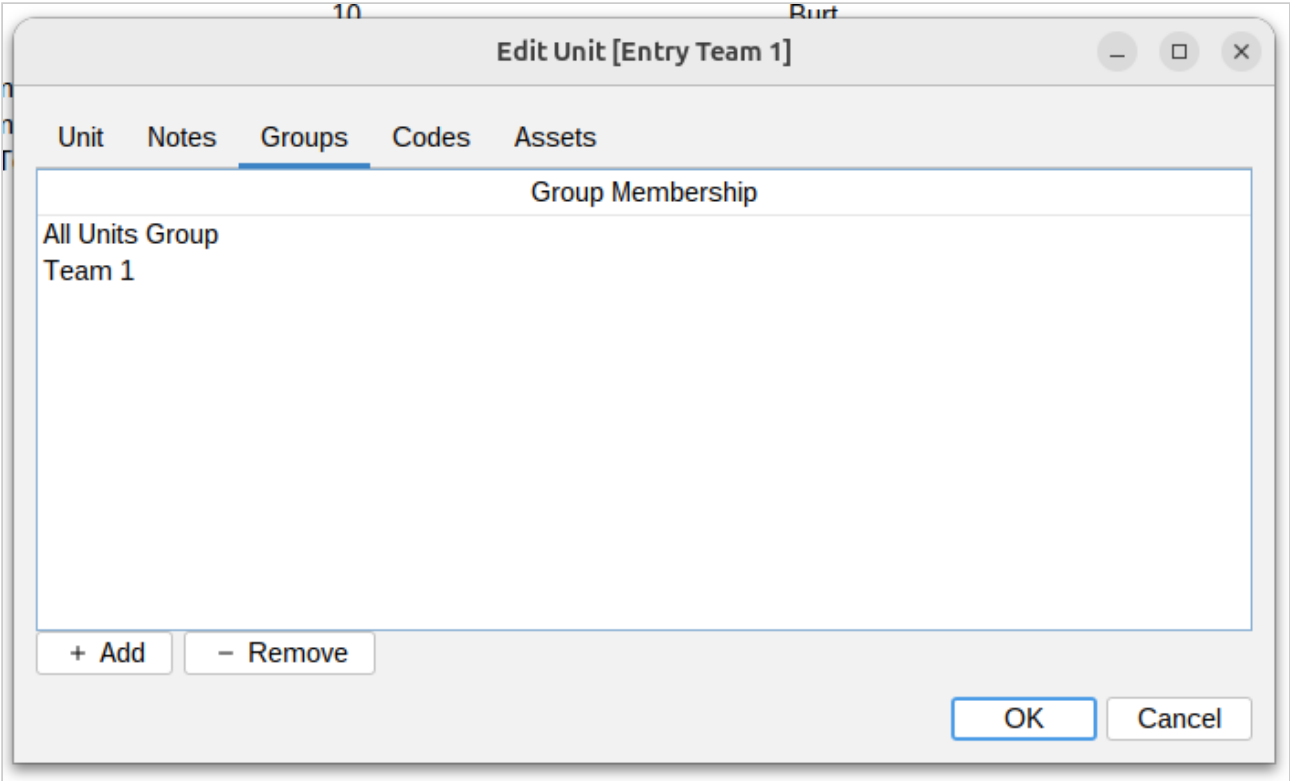
The labels for Field A, Field B, and Field C are set in the Units Settings panel under Options. Rename these fields to match the type of information you want to track for each unit in your communications plan.

The Name field is the only required field. All other fields are optional. When you click OK without entering a name, a validation error is displayed and focus returns to the Name field.

Related Topics

- Unit Edit Dialog
- Units Settings -- configure custom field labels and auto-numbering
- Units Panel

Unit Groups Tab



The Groups tab in the Unit Edit Dialog shows which groups this unit belongs to. Groups organize units into logical sets that share common cryptographic material such as callsigns, authentication codes, and passwords. A unit can belong to multiple groups simultaneously.

Table Columns

Column	Description
Group Membership	The name of each group this unit belongs to.

Actions

- **Add** -- Click the **+** button to open a group selector dialog. Select one or more groups to add this unit to. Groups the unit already belongs to are excluded from the selector.
- **Remove** -- Select one or more groups in the table and click the **−** button to remove the unit from those groups.

> Every unit is automatically a member of the **All Units** group. This group cannot be removed -- it ensures that every unit receives at least the base set of cryptographic material defined at the CEOI level.

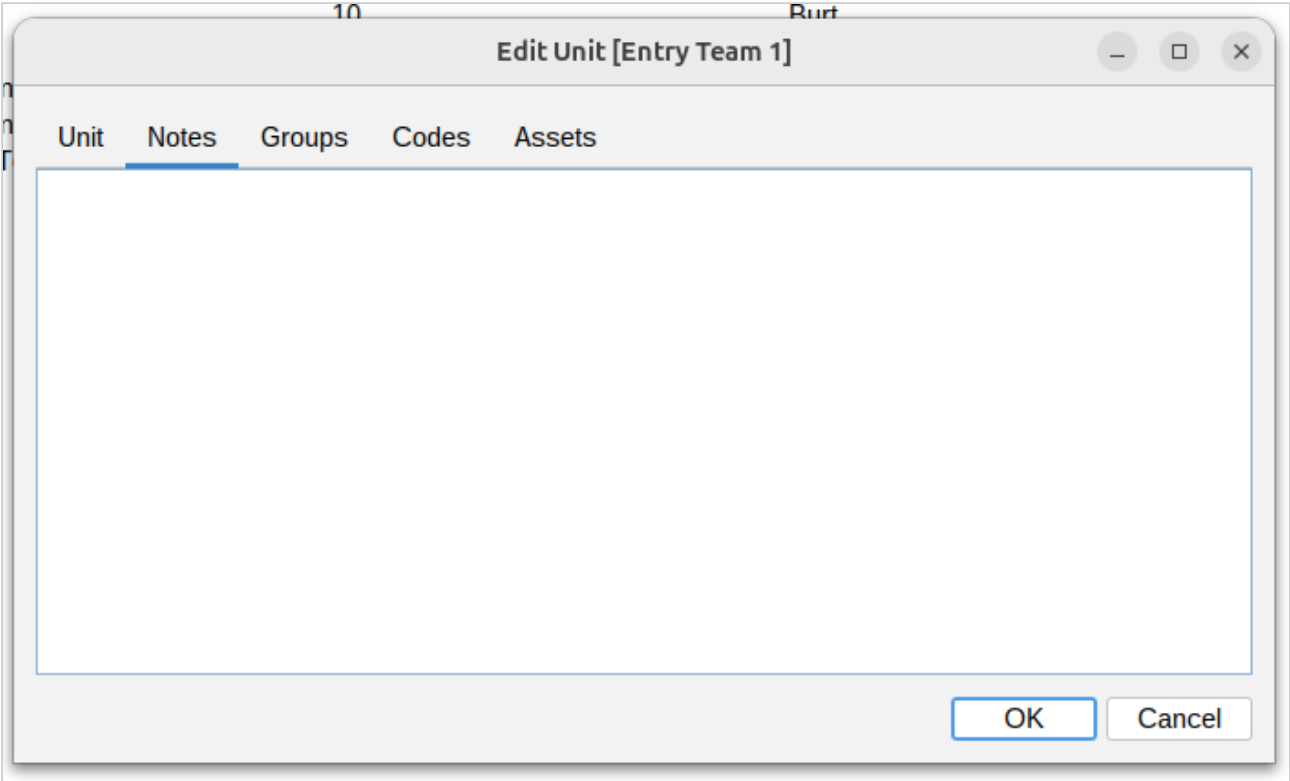
> Adding a unit to additional groups allows it to receive separate callsigns, authentication codes, and passwords for each group. This is useful when a unit participates in multiple nets or operational

> contexts. You can view and manage the codes for each group on the Codes tab by selecting the group from the Group dropdown.

Related Topics

- Unit Edit Dialog
- Codes Tab -- view codes per group
- Units Panel

Unit Notes Tab



The Notes tab in the Unit Edit Dialog provides a free-text area where you can record notes about the unit. Notes are saved with the unit and can contain any information relevant to your communications plan -- operational remarks, equipment details, special instructions, or any other text.

Usage

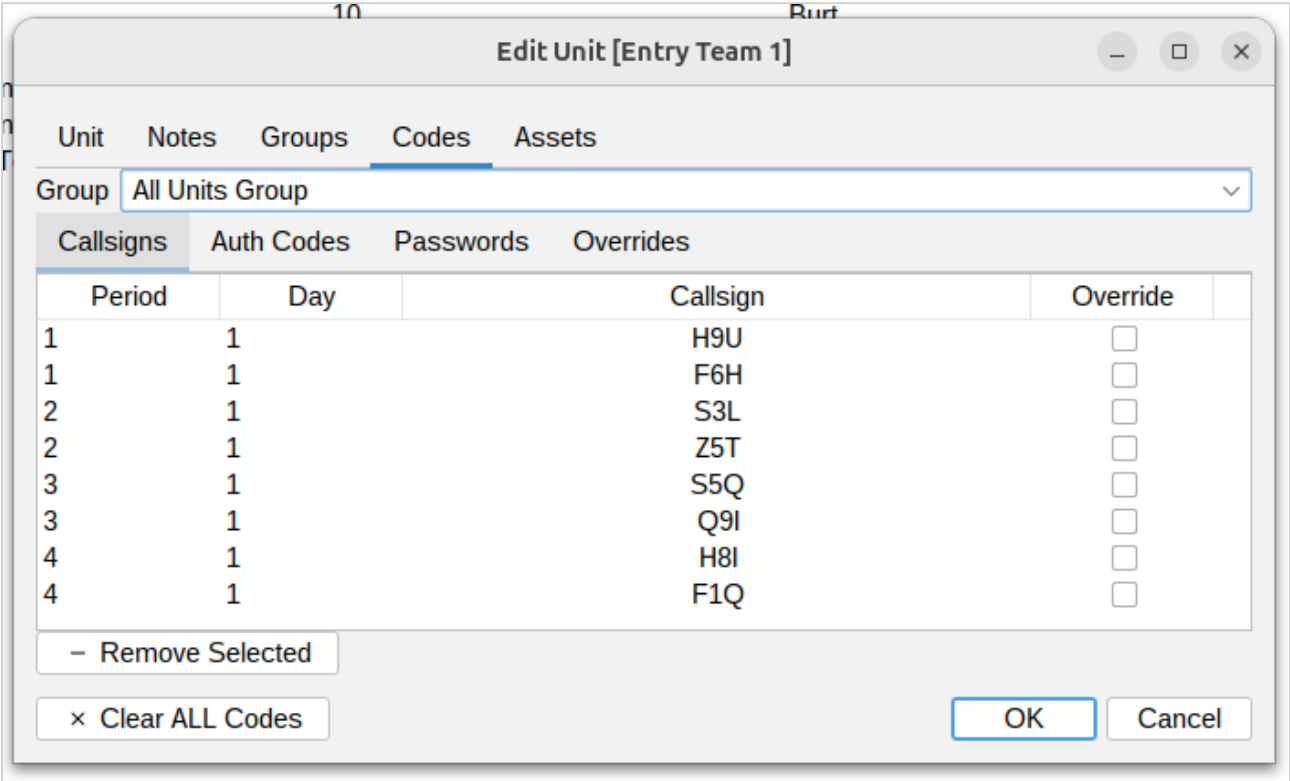
Type or paste text directly into the text area. There is no character limit. The content is saved when you click **OK** in the dialog.

> Notes are stored as plain text. They are included when the unit is printed, so you can use this field to add context that will appear on the printed CEOI page for this unit.

Related Topics

- Unit Edit Dialog
- Unit Tab

Unit Callsigns Tab



The Callsigns tab is a sub-tab within the Codes tab of the Unit Edit Dialog. It displays the callsigns assigned to this unit for the currently selected group. Callsigns are generated during crypto generation and identify the unit in radio communications.

Group Selector

The **Group** dropdown at the top of the Codes tab controls which group's codes are displayed. If the unit belongs to multiple groups, select a group from the dropdown to view that group's callsign assignments. Each group membership has its own independent set of callsigns.

Table Columns

Column	Description
Period	The CEOI period number for which this callsign is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Callsign	The assigned callsign code. This column is editable -- you can click a cell to type a replacement callsign. Manually edited callsigns are marked as overrides.
Override	A checkbox indicating whether this callsign has been manually overridden. When you edit a callsign value directly, the override flag is set to true automatically.

Editing Callsigns

You can manually override any generated callsign by clicking directly on the callsign value in the table and typing a new value. The entered value is automatically converted to uppercase. If the callsign you enter is already assigned to another unit for the same period and division, a duplicate warning is displayed and the change is rejected.

Actions

- **Remove Selected** -- Select one or more rows in the table and click **Remove Selected** to delete those callsign assignments.

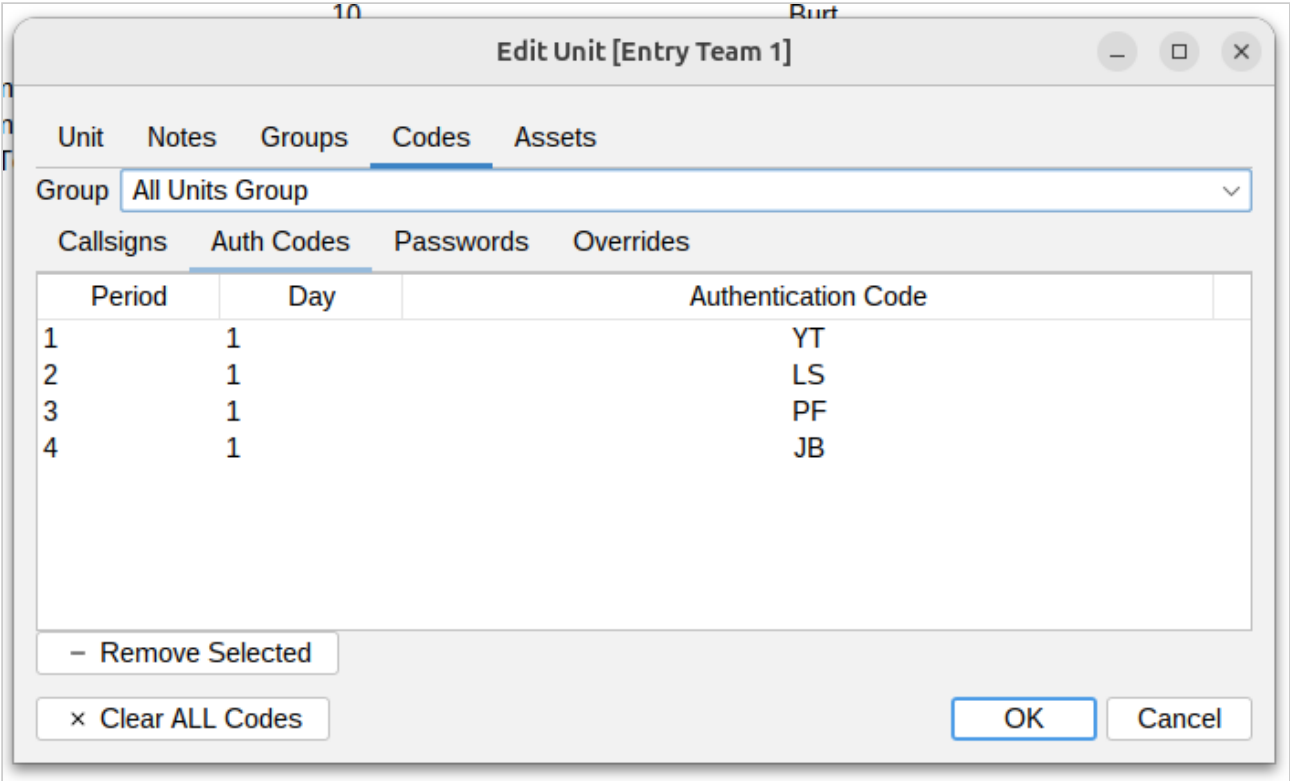
> Removing callsigns or overriding them with custom values affects the generated crypto. Overridden callsigns are kept when crypto is regenerated; if the callsign count is lowered, generated callsigns are removed first and overrides last. Use Clear CEOI Crypto to remove overrides.

> You can sort the callsigns table by clicking on any column header. This is useful for quickly finding callsigns for a specific period or division.

Related Topics

- Codes Tab
- Unit Edit Dialog
- Auth Codes Tab
- Passwords Tab

Unit Auth Codes Tab



The Auth Codes tab is a sub-tab within the Codes tab of the Unit Edit Dialog. It displays the authentication codes assigned to this unit for the currently selected group. Authentication codes are used to verify the identity of a communicating party during radio exchanges.

Group Selector

The **Group** dropdown at the top of the Codes tab controls which group's codes are displayed. If the unit belongs to multiple groups, select a group from the dropdown to view that group's authentication code assignments.

Table Columns

Column	Description
Period	The CEOI period number for which this authentication code is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Authentication Code	The generated authentication code value.

Actions

- **Remove Selected** -- Select one or more rows in the table and click **Remove Selected** to delete those authentication code assignments.

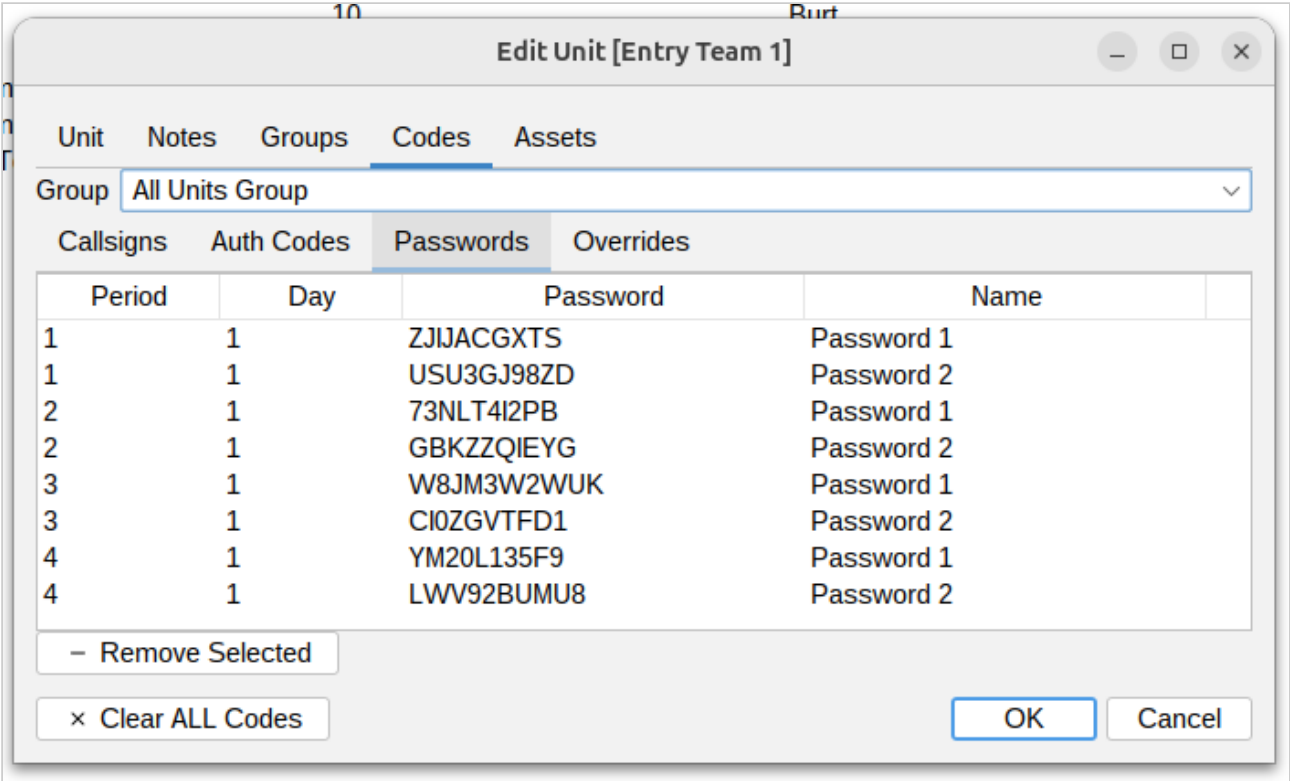
> Authentication codes are read-only in this table -- unlike callsigns, they cannot be edited inline. To change authentication code generation settings, use the Overrides sub-tab or the Units Settings panel.

> The number of authentication codes generated per unit per group is controlled by the Auth Code Count on the Overrides sub-tab. If no override is set, the CEOI-level default from Units Settings is used.

Related Topics

- Codes Tab
- Unit Edit Dialog
- Callsigns Tab
- Passwords Tab

Unit Passwords Tab



The Passwords tab is a sub-tab within the Codes tab of the Unit Edit Dialog. It displays the passwords assigned to this unit for the currently selected group. Passwords are generated during crypto generation and are used for authentication or access control in the communications plan.

Group Selector

The **Group** dropdown at the top of the Codes tab controls which group's codes are displayed. If the unit belongs to multiple groups, select a group from the dropdown to view that group's password assignments.

Table Columns

Column	Description
Period	The CEOI period number for which this password is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Password	The generated password value.
Name	The descriptive name assigned to this password slot. Password names are configured in the CEOI password settings and provide a human-readable label for each password (for example, "Primary" or "Alternate").

Actions

- **Remove Selected** -- Select one or more rows in the table and click **Remove Selected** to delete those password assignments.

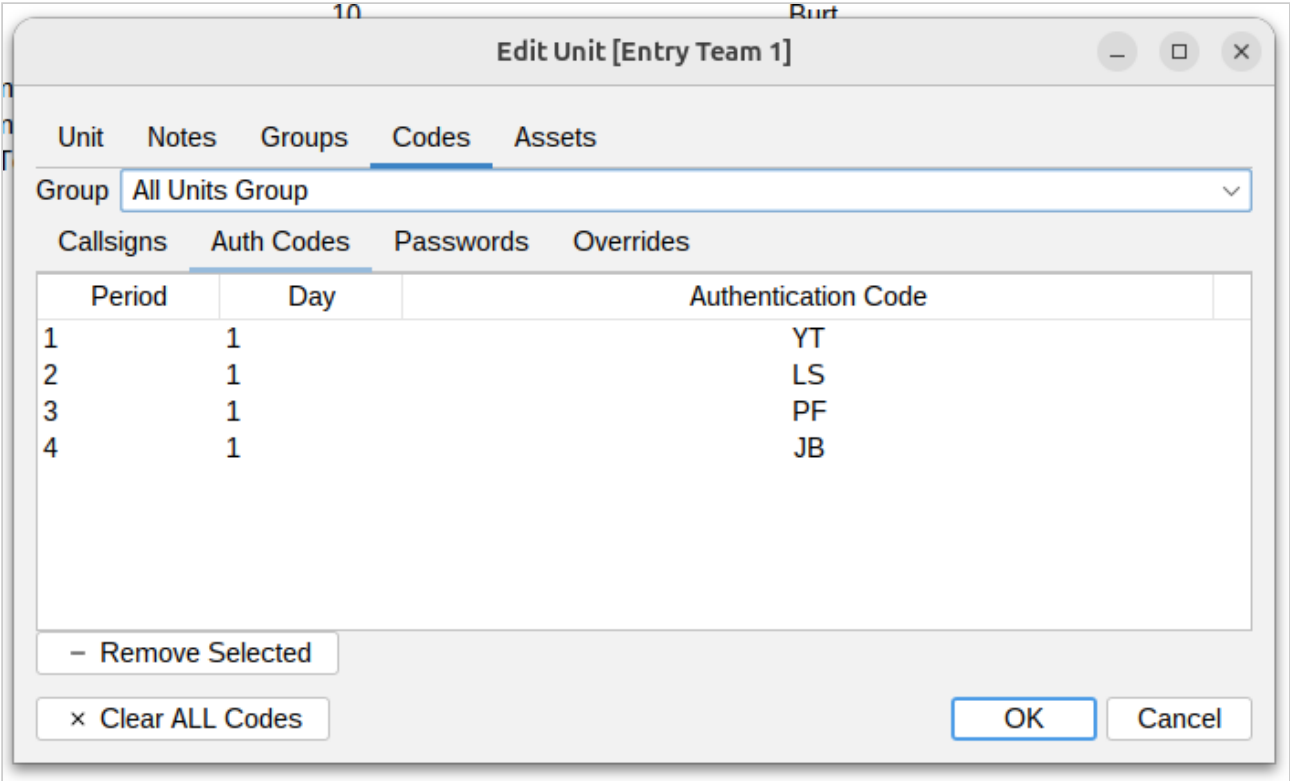
> Passwords are read-only in this table -- they cannot be edited inline. To change password generation settings such as length, character set, or count, use the Overrides sub-tab or the Units Settings panel.

> The number of passwords generated per unit per group is controlled by the Password Count on the Overrides sub-tab. If no override is set, the CEOI-level default from Units Settings is used.

Related Topics

- Codes Tab
- Unit Edit Dialog
- Callsigns Tab
- Auth Codes Tab

Unit Code Overrides



The **Overrides** tab sits inside the Codes tab of the Unit Edit Dialog. It sets how many of each code this one unit is given per division, overriding the CEOI-wide defaults from Unit Settings.

Field	Description
Callsign Count	How many call signs to generate for this unit in each division.
Auth Code Count	How many authentication codes to generate for this unit in each division.
Password Count	How many passwords to generate for this unit in each division.

> Leave a field empty to use the CEOI default. A value here applies to this unit only, and takes effect the next time codes are generated -- changing it does not alter codes already generated.

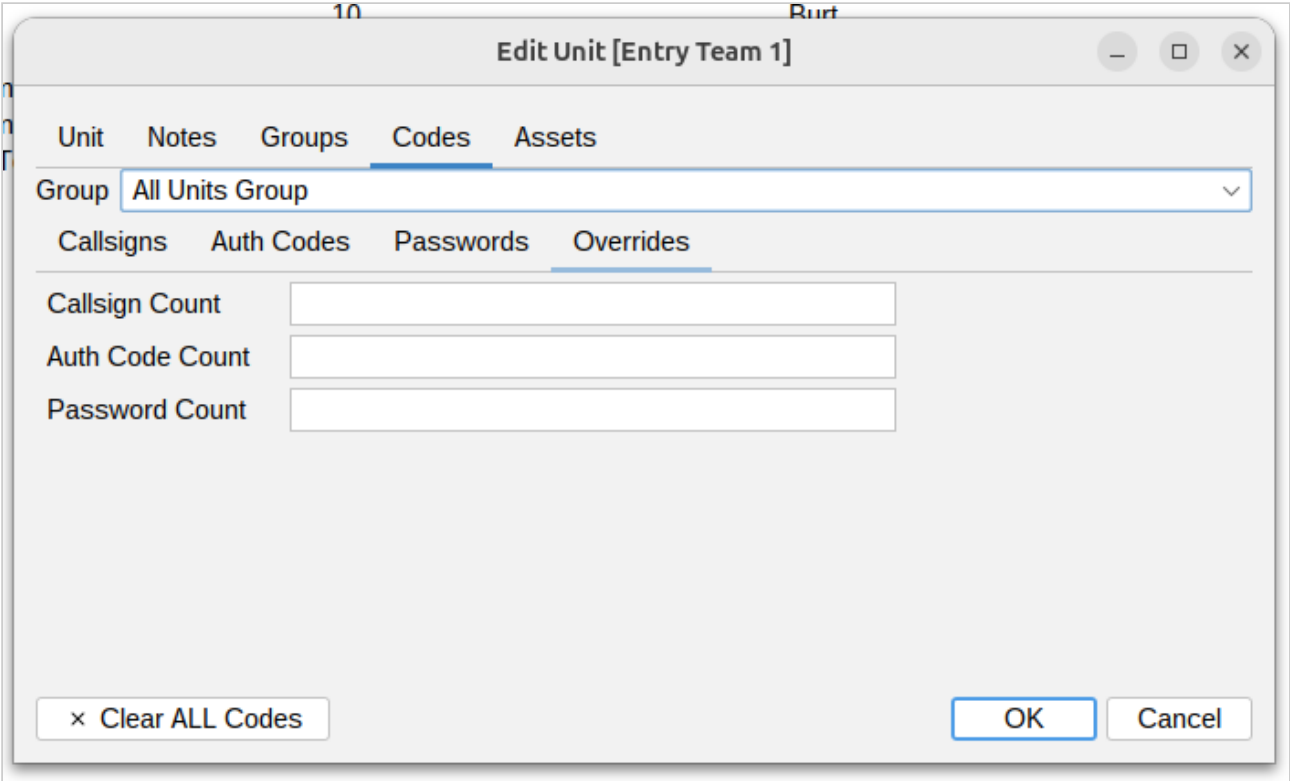
> Giving one unit more call signs than the rest is a way to let a busy net control station rotate through several in a single division without reusing one.

Related Topics

- Unit Edit Dialog
- Unit Codes Tab
- Call Signs Tab
- Auth Codes Tab
- Passwords Tab

- Unit Settings

Unit Codes Tab



The Codes tab in the Unit Edit Dialog displays all generated cryptographic codes for the unit, organized by group membership. This tab contains a group selector and a set of sub-tabs for each code type.

Group Selector

The **Group** dropdown at the top of the tab lets you choose which group's codes to view. Since a unit can belong to multiple groups (see the Groups tab), each group membership maintains its own independent set of callsigns, authentication codes, and passwords. Selecting a different group updates all the sub-tabs below to show that group's codes.

Sub-Tabs

Sub-Tab	Description
Callsigns	Lists the callsigns assigned to this unit for the selected group. Callsigns can be manually overridden by editing them directly in the table.
Auth Codes	Lists the authentication codes assigned to this unit for the selected group.
Passwords	Lists the passwords assigned to this unit for the selected group, along with their assigned names.
Overrides	

	Allows you to set per-unit, per-group overrides for the number of callsigns, authentication codes, and passwords generated. When left blank, the CEOI-level defaults from Units Settings are used.

Overrides Sub-Tab

The Overrides sub-tab provides fields to override the default code counts for this specific unit within the selected group:

Field	Description
Callsign Count	The number of callsigns to generate for this unit in the selected group. Valid values range from 1 to 99. Leave blank to use the CEOI default.
Auth Code Count	The number of authentication codes to generate for this unit in the selected group. Valid values range from 1 to 99. Leave blank to use the CEOI default.
Password Count	The number of passwords to generate for this unit in the selected group. Valid values range from 1 to 99. Leave blank to use the CEOI default.

Clear ALL Codes

When the Codes tab is selected, a **Clear ALL Codes** button appears in the dialog's button bar. Clicking this button removes all callsigns, authentication codes, and passwords for the currently selected group membership after a confirmation prompt. This action cannot be undone within the dialog -- you would need to cancel the dialog to discard the change, or regenerate crypto after saving.

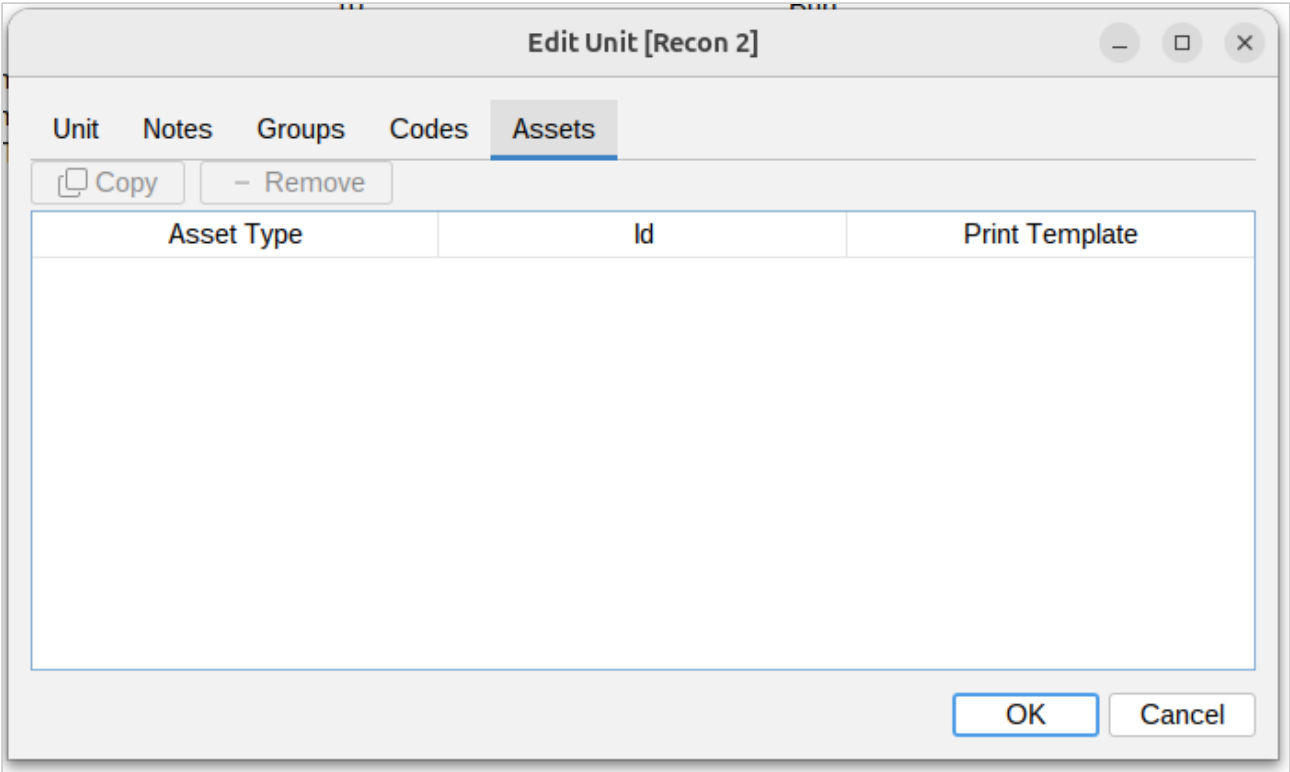
> Clearing all codes removes every callsign, authentication code, and password for the selected group. The codes will not be restored unless you regenerate crypto for the CEOI.

> Use the Overrides sub-tab to customize code counts for specific units that need more or fewer codes than the CEOI default. For example, a headquarters unit might need additional callsigns for its multiple radio stations, while a simple relay station might need only one.

Related Topics

- Unit Edit Dialog
- Groups Tab -- manage group memberships
- Callsigns Tab
- Auth Codes Tab
- Passwords Tab
- Units Settings -- CEOI-level defaults

Unit Assets Tab



The Assets tab in the Unit Edit Dialog displays the cryptographic and communications assets that have been assigned to this unit. Assets are shared resources -- such as DRYADs, SARNEGs, One-Time Pads, Challenge/Response tables, and SOIs -- that are assigned to units and groups through the asset assignment system.

Table Columns

Column	Description
Asset Type	The category of the asset. Possible types include DRYAD, One Time Pad, SARNEG, Challenge/Response, SOI, Units, and Groups.
Id	The identifying name or label of the specific asset. For unit and group asset types, this column shows "Member Units" or "Member Groups" respectively.
Print Template	The print template assigned to this asset for printed output. Click this cell to open a dropdown and select from the available print templates that are compatible with the asset type. This controls how the asset appears when the unit's CEOI materials are printed.

Interactions

- **Double-click** -- Double-click a row to open a view dialog for that asset, allowing you to inspect its full details.

- **Print Template selection** -- Click the Print Template cell for any row to choose which template is used when printing this asset for the unit.

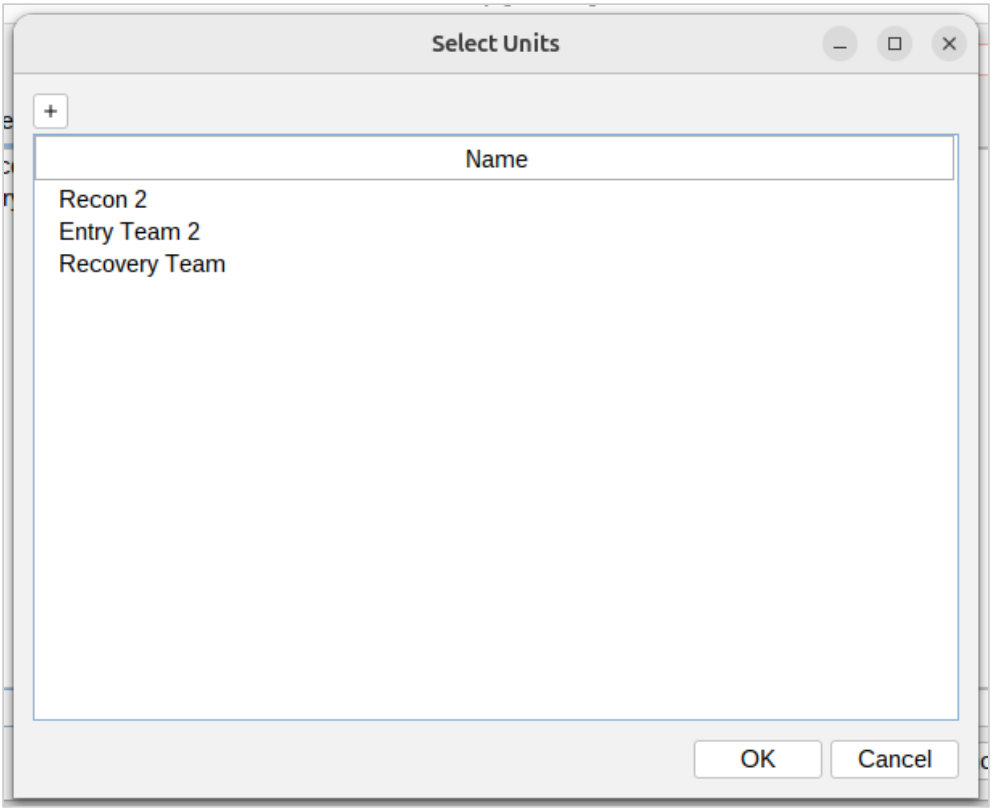
> The Assets tab is read-only with respect to which assets are assigned. Assets are assigned to units from the asset's own editor -- for example, from the DRYAD, OTP, or SARNEG panels in the main CEOI editor using the Asset Assignment Dialog. The Assets tab here shows you which assets this unit has received.

> Assigning a print template to each asset controls which layout is used when generating printed CEOI materials for this unit. If no template is selected, the default template for that asset type is used.

Related Topics

- Unit Edit Dialog
- Asset Assignment Dialog -- assign assets to units and groups
- Asset Assignment Panel -- manage assignments within asset editors
- Groups Tab -- manage group membership

Unit Selector



The Unit Selector dialog allows you to choose one or more units from the CEOI. It is used throughout Scepter whenever you need to pick units -- for example, when assigning units to an asset through the Asset Assignment Panel.

Table Columns

The dialog displays a table listing all available units in the current CEOI. Each row shows a unit with a checkbox for selection. Units that are already assigned or otherwise excluded from the current context are not shown in the list.

Column	Description
Checkbox	Check the box next to each unit you want to select. You can select multiple units at once.
Unit Name	The name of the unit as defined in the CEOI.

Actions

Action	Description
New (+)	Click the **+** button at the top of the dialog to create a new unit. The Unit Edit Dialog opens, and after you save the new unit it appears in the selector list.

OK	Confirms your selection and returns the checked units to the calling dialog.
Cancel	Closes the dialog without making a selection. You can also press `Escape` to cancel.

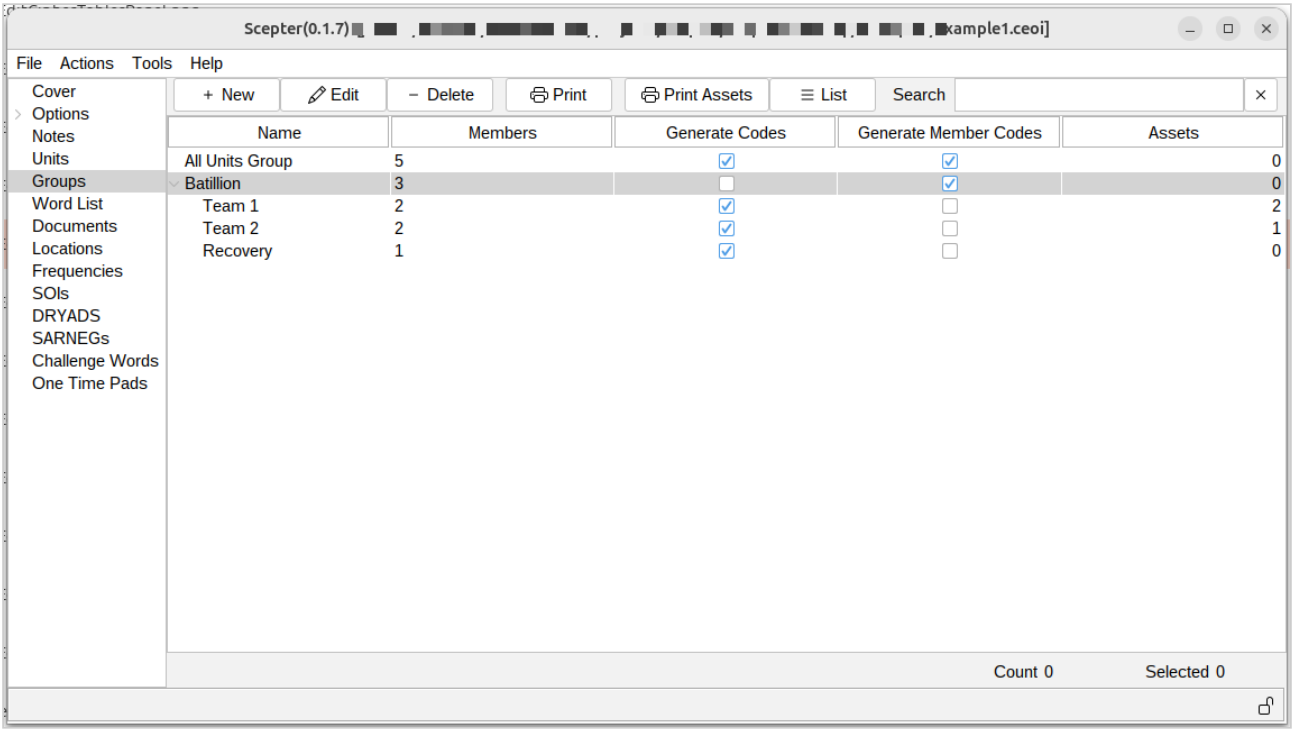
> The table supports sorting. Click a column header to sort the units alphabetically, making it easier to find a specific unit in a large CEOI.

> If the unit you need does not exist yet, use the **+** button to create it directly from this dialog. You do not need to close the selector, create the unit separately, and then return.

Related Topics

- Unit Edit Dialog -- create and edit units
- Asset Assignment Panel -- uses the unit selector for asset assignments
- Entity Selector -- generic entity selection dialog
- Units Panel

Groups



The Groups panel displays all communication groups (nets) defined in the current CEOI. Groups organize units into logical communication networks. A group contains member units and can also contain other groups, forming a hierarchy.

View Modes

The panel supports two view modes, toggled with the view button in the toolbar:

- **Tree View** – Displays groups in a hierarchical tree. Top-level groups appear as root nodes, and sub-groups are nested beneath their parent groups. This is the default view.
- **List View** – Displays all groups in a flat list, regardless of their hierarchy. The list view is activated automatically when you type in the search field.

Actions

- **New** – Click the **+** button or press `Insert` to create a new group. The Group Edit Dialog opens.
- **Edit** – Select a group and click the pencil button, press `Enter`, or double-click to open the Group Edit Dialog.
- **Delete** – Select one or more groups and click the **−** button or press `Delete`. Deleting a group removes it from any parent groups and detaches its member units.
- **Print** – Select one or more groups and click the printer button to print CEOI pages for the selected groups.
- **Print Assets** – Select a group and click Print Assets to open the Group Asset Print Dialog, which prints a complete CEOI for the group's members.

Search

Type in the search field to filter groups by name. When filtering is active, the view automatically switches to list mode. Clear the search field to return to the full list.

Context Menu

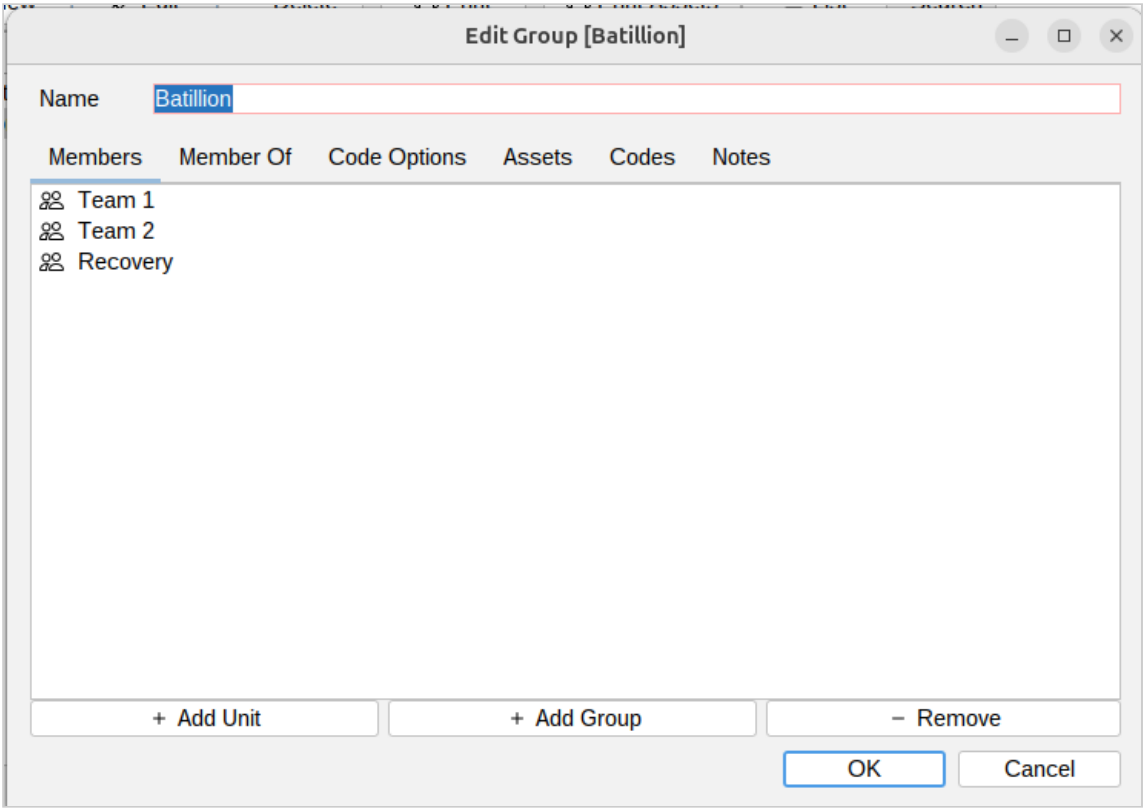
Right-click anywhere in the panel to access the context menu with options to add, edit, delete, print, print assets, import, and export groups.

Import and Export

Use the context menu or **Tools > Import List / Export List** to import or export group data.

> The status bar at the bottom shows the total group count and how many are currently selected.

Group Edit Dialog



The Group Edit Dialog opens when you add or edit a group from the Groups panel. It lets you configure the group's name, members, code generation options, and view generated codes.

Name

The **Name** field at the top of the dialog is required. It is highlighted with a pink border to indicate it cannot be left blank.

Tabs

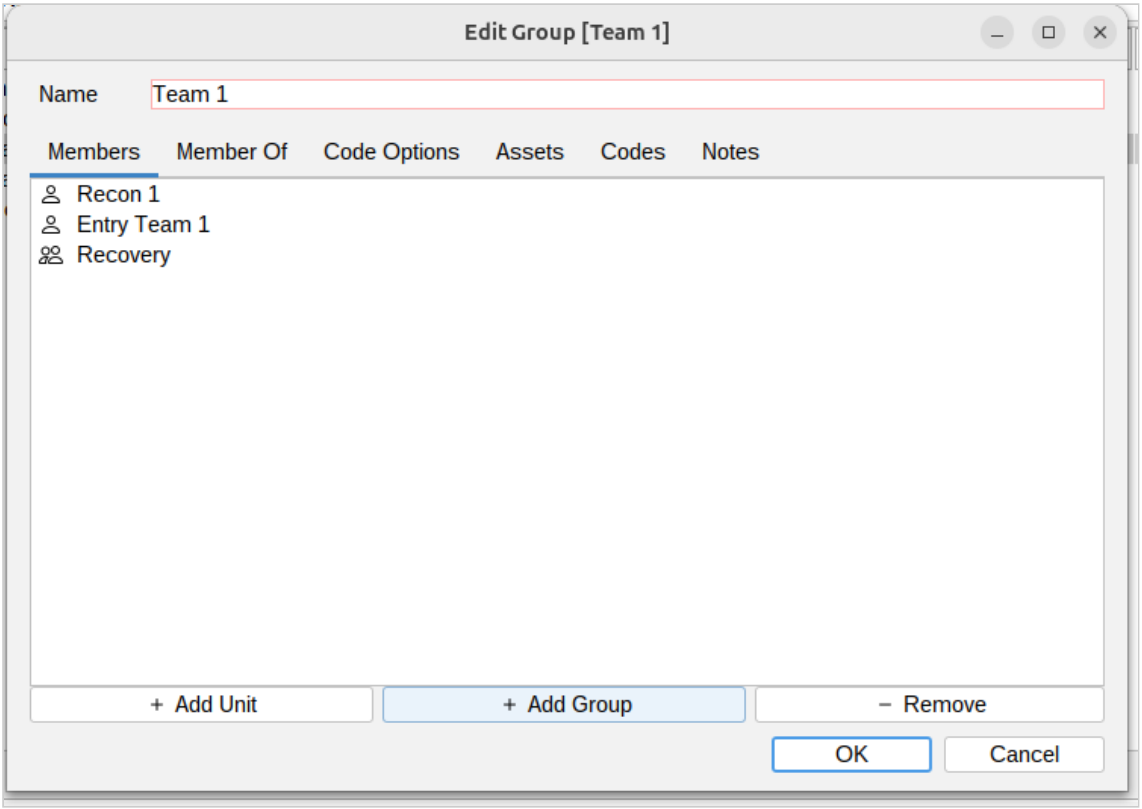
Tab	Description
Members	Add and remove units and sub-groups that belong to this group.
Member Of	View which parent groups this group belongs to (read-only).
Options	Configure code generation settings for members of this group, including callsign patterns, auth codes, and passwords.
Assets	View assets assigned to this group.
Codes	View generated codes (callsigns, auth codes, passwords) for this group's members, with per-group override settings.
Notes	Free-text notes about this group.

Buttons

- **OK** – Validate the group name, save all changes, and close the dialog. Press `Enter` as a shortcut.
- **Cancel** – Discard all changes and close the dialog. Press `Escape` as a shortcut.

> The built-in "All Units" group cannot have its name changed, and you cannot add or remove members from it.

Group Members



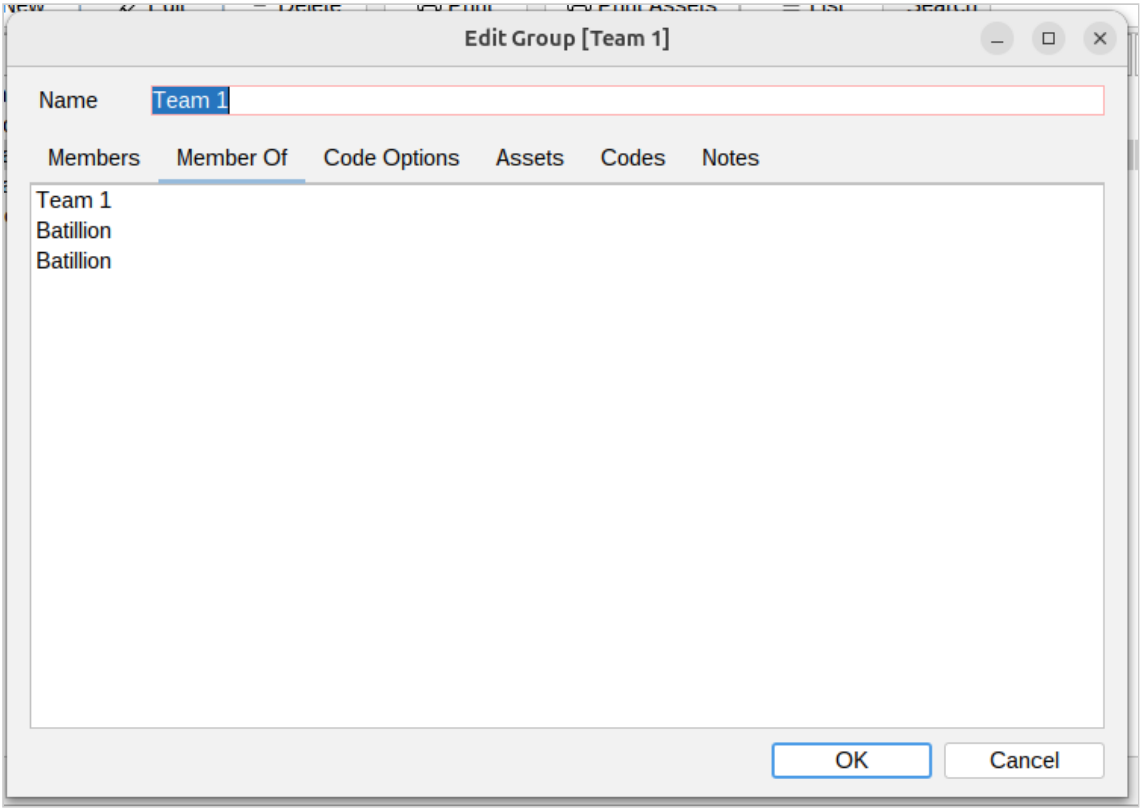
The Members tab in the Group Edit Dialog shows which units and sub-groups belong to this group. Each member is listed with an icon indicating whether it is a unit (person icon) or a group (people icon).

Actions

- **Add Unit** – Opens the Unit Selector to choose units to add as members. Units already in the group are excluded from the selection list.
- **Add Group** – Opens the Group Selector to choose sub-groups to add as members. The current group, the "All Units" group, and groups already included are excluded from the selection list.
- **Remove** – Select one or more rows and click Remove to detach them from this group.

> A group can contain both individual units and other groups, allowing you to build hierarchical communication networks.

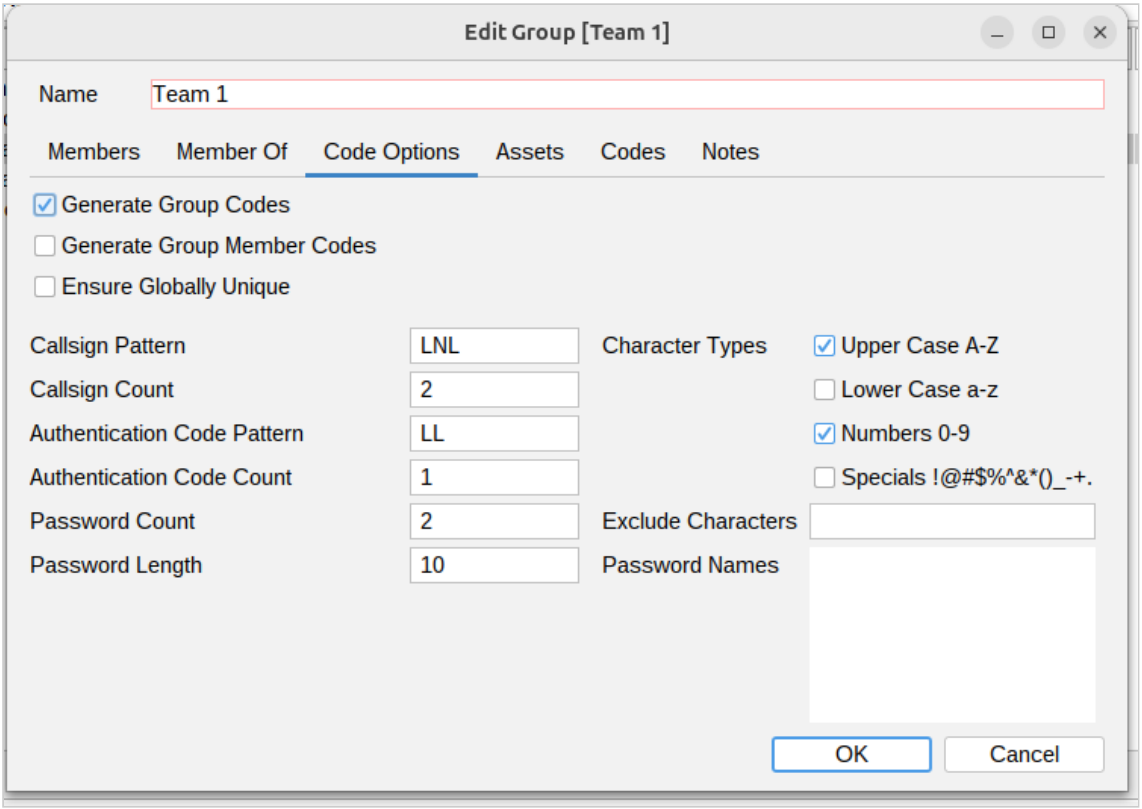
Group Member Of



The Member Of tab in the Group Edit Dialog shows a read-only list of all parent groups that this group belongs to. Every group is always a member of itself (shown as the first entry).

To change which parent groups this group belongs to, edit the parent group and add or remove this group from its Members tab.

Group Options



The Options tab in the Group Edit Dialog controls code generation settings for the members of this group. These settings override the global defaults from Units Settings when codes are generated for units within this group context.

Generation Checkboxes

Option	Description
Generate Codes	Enable code generation for this group's members.
Generate for Members	Generate individual codes for each member unit within this group.
Globally Unique	Ensure generated codes are unique across all groups, not just within this group.

Callsign Settings

Field	Description
Callsign Pattern	The code pattern for generating callsigns. See Generated Code Patterns.
Callsign Count	Number of callsigns to generate per unit per division (0–100).

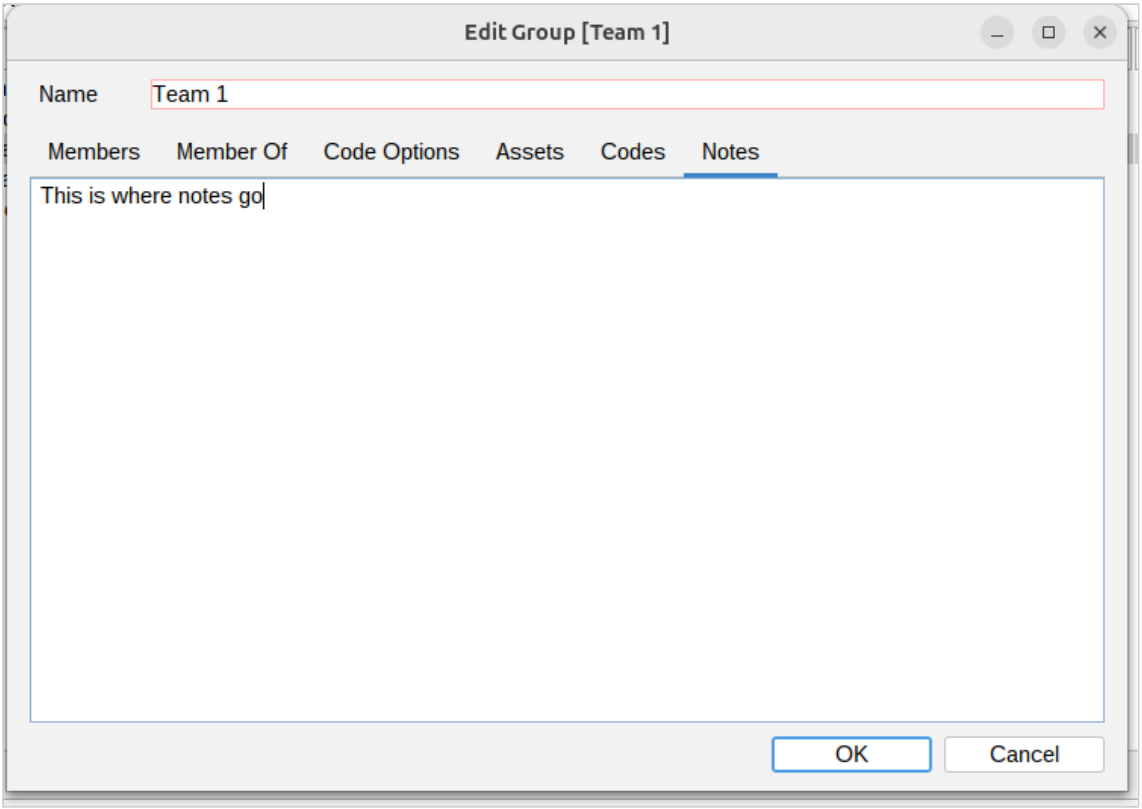
Authentication Code Settings

Field	Description
Authentication Code Pattern	The code pattern for generating auth codes.
Authentication Code Count	Number of auth codes to generate per unit per division (0–100).

Password Settings

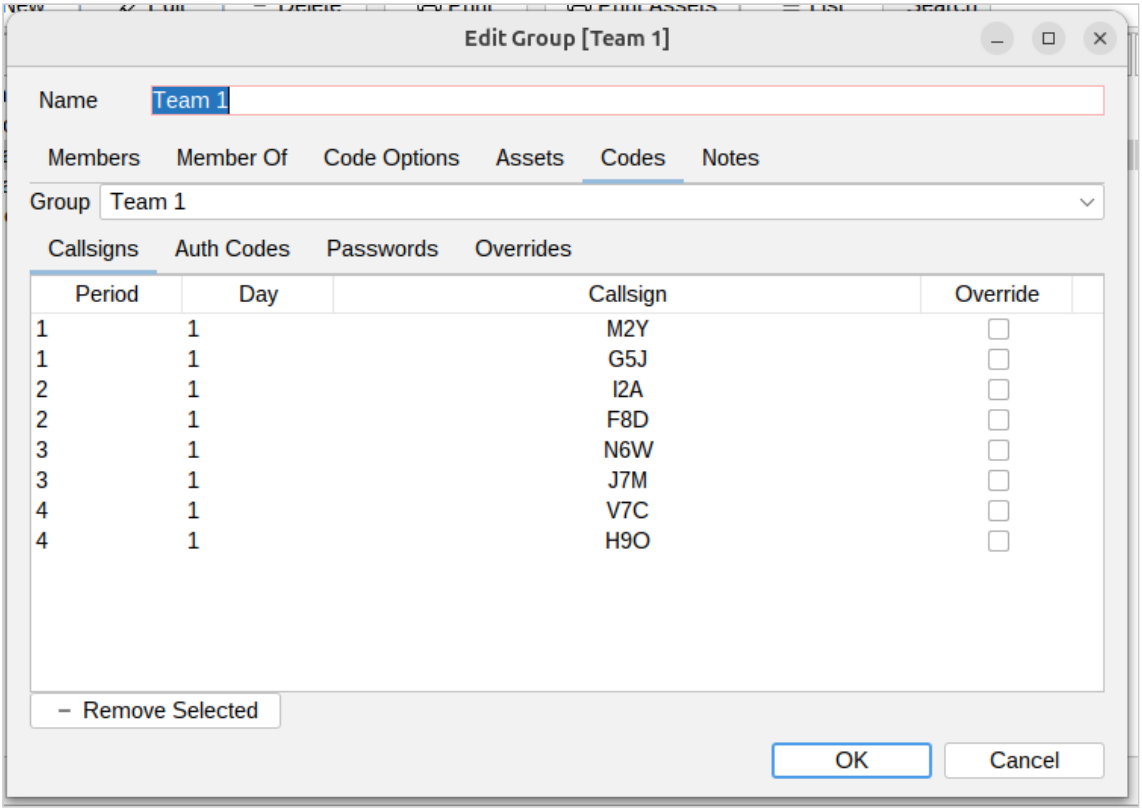
Field	Description
Password Count	Number of passwords to generate per unit per division (0–100).
Password Length	The character length of each generated password (0–15).
Character Types	Select which character sets to include: Upper Case (A–Z), Lower Case (a–z), Numbers (0–9), Specials (!@#$%^&*()_-+.).
Exclude Characters	Characters to exclude from passwords (e.g., "oO" to avoid ambiguous characters).
Password Names	Optional names for each password position (e.g., "Primary", "Alternate").

Group Notes



The Notes tab in the Group Edit Dialog provides a free-text area where you can record notes about the group. Notes are saved with the group and included in the CEOI file.

Group Codes



The Codes tab in the Group Edit Dialog displays the generated cryptographic codes for this group's members. A dropdown at the top lets you select which parent group context to view codes for.

Group Selector

The **Group** dropdown at the top lists the groups this group is a member of. Selecting a different group shows the codes generated for that group context.

Code Sub-tabs

Sub-tab	Description
Callsigns	Displays callsigns with columns for Period, Division, Callsign, and Override flag. Callsigns can be edited inline — changes are automatically uppercased and checked for duplicates.
Auth Codes	Displays authentication codes with columns for Period, Division, and Authentication Code.
Passwords	Displays passwords with columns for Period, Division, Password, and Name.
Overrides	Override the code counts for this specific group membership. Fields for Callsign Count, Auth Code Count, and Password Count let you customize how many codes are generated for this group context.

Actions

Each code sub-tab has a **Remove Selected** button to delete selected generated codes.

> Override counts on the Overrides sub-tab apply only to this group's membership context. Leave fields blank to use the defaults from the Options tab.

Group Assets

The Assets tab in the Group Edit Dialog displays the cryptographic and communications assets issued to this group — DRYADs, One-Time Pads, SARNEGs, Challenge/Response tables and SOIs. Every member of the group is issued what the group holds. It is the same view the Unit Assets Tab gives for a unit.

Table Columns

Column	Description
Asset Type	The category of the asset: DRYAD, One Time Pad, SARNEG, Challenge/Response, SOI, Units or Groups.
Id	The identifying name or label of the specific asset. For the Units and Groups types this reads “Member Units” or “Member Groups” — the group's own roster, printed as part of its packet.
Print Template	The print template used for this asset when the group's materials are printed. Click the cell to choose from the templates compatible with the asset type. If none is selected, the default template for that asset type is used.

Interactions

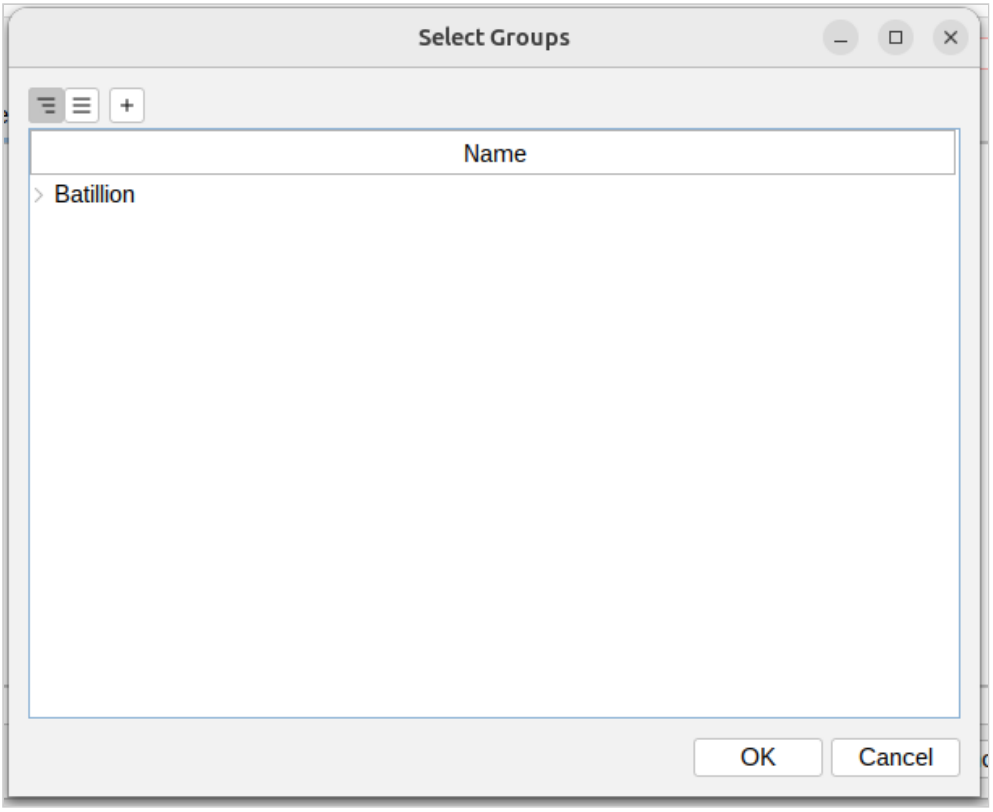
- **Double-click** -- Open a view dialog for that asset to inspect its full details.
- **Print Template selection** -- Click the Print Template cell on any row to choose the layout used when printing this asset for the group.

> This tab is read-only with respect to *which* assets are held. Assets are issued to a group from the asset's own editor -- the DRYAD, OTP, SARNEG or Challenge/Response panels in the main CEOI editor, using the Asset Assignment Dialog and its **Groups** button. This tab shows the result and controls how each asset prints.

Related Topics

- Group Edit Dialog
- Group Members Tab
- Asset Assignment Dialog
- Unit Assets Tab
- SOI Assets Tab
- Group Asset Print

Group Selector



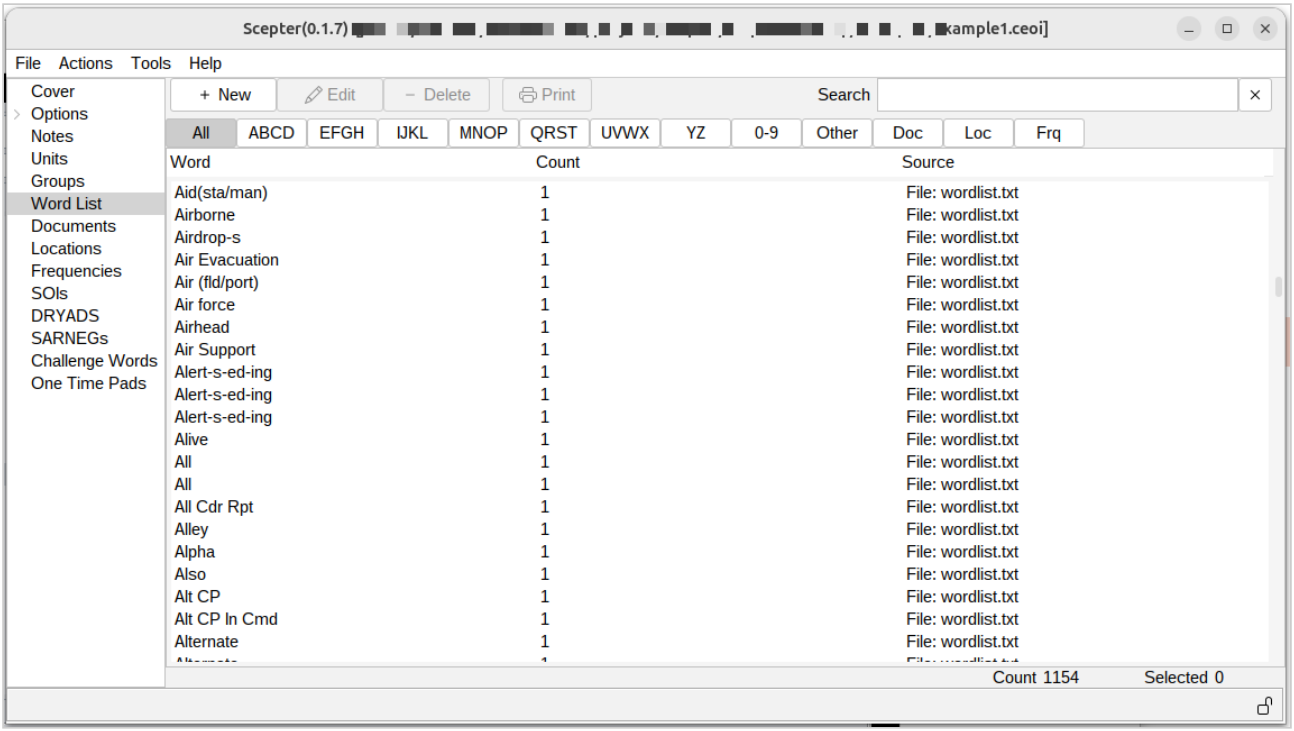
The Group Selector dialog lets you choose one or more groups from a list. It appears when you need to select groups, such as when adding sub-groups to a group's members.

Usage

- Check the boxes next to the groups you want to select.
- Click **OK** to confirm your selection.
- Click **Cancel** or press `Escape` to close without selecting.

> Groups that are already members of the target group and the "All Units" group are excluded from the selection list.

Word List



The Word List panel displays all code words defined in the current CEOI. Code words are used for encoding and decoding messages using the Word List Encode/Decode tool. Words can be entered manually or linked automatically from frequencies, locations, and documents.

Table Columns

Column	Description
Word	The code word text.
Count	The number of code assignments for this word.
Source	Where this word originated (manually added, or linked from a frequency, location, or document).

Filter Bar

The word list includes a filter bar that lets you quickly narrow the displayed words:

- **All** – Show all words.
- **A–Z** – Click a letter to show only words starting with that letter.
- **Numbers** – Show words starting with a digit.
- **Other** – Show words starting with a non-alphanumeric character.
- **Documents** – Show only words linked from documents.
- **Locations** – Show only words linked from locations.

- **Frequencies** – Show only words linked from frequencies.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new word. The Word Edit Dialog opens.
- **Edit** – Select a word and click the pencil button, press `Enter`, or double-click to edit it.
- **Delete** – Select one or more words and click the **−** button or press `Delete`.

Search

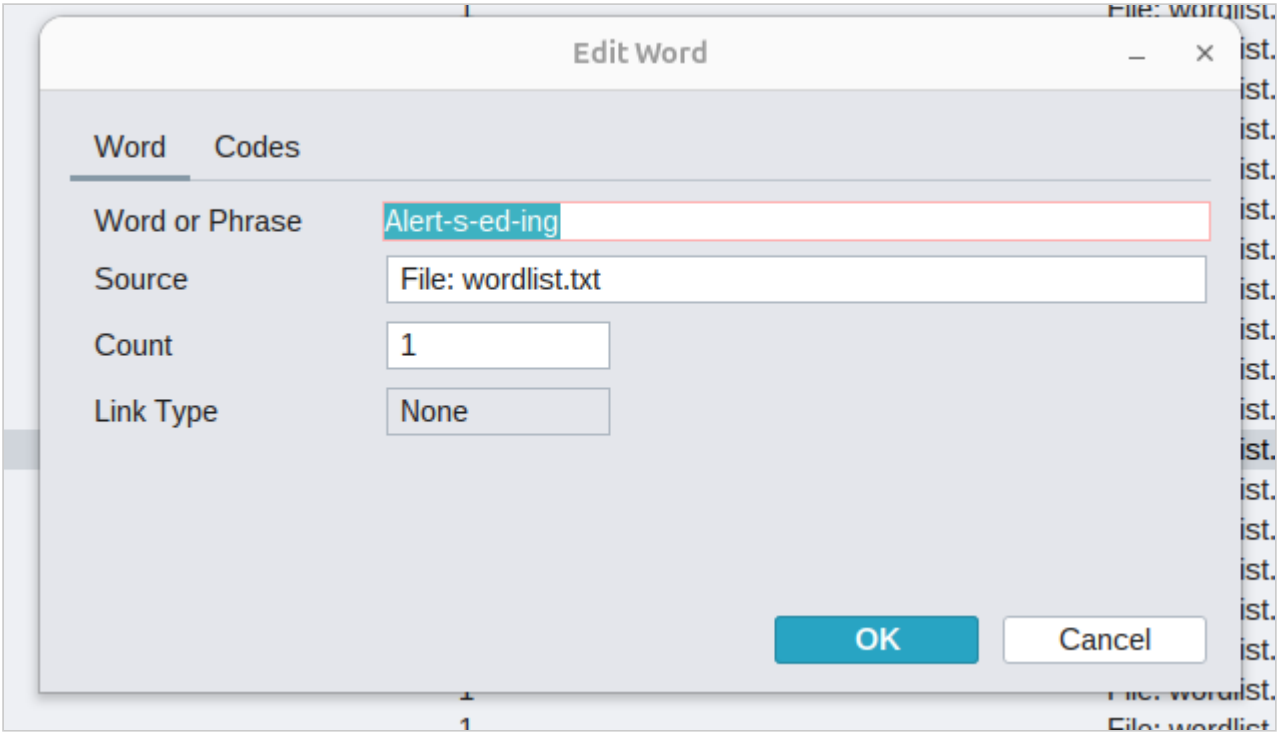
Type in the search field to filter words by name. Clear the search to show all words.

Import and Export

Use **Tools > Import List** and **Tools > Export List** (or the right-click context menu) to import or export word list data.

> Words that are linked from frequencies, locations, or documents are managed through their respective entity. Editing or deleting the source entity will update or remove the linked word automatically.

Word Edit Dialog



The Word Edit Dialog is used to create or edit a word entry in the CEOI word list. It is a tabbed dialog with tabs for editing word properties, notes, and viewing generated codes.

Tabs

Tab	Description
Word	Edit the word or phrase and its properties. See Word Edit Panel.
Notes	Free-text notes about this word.
Codes	View generated codes assigned to this word. See Word Codes.

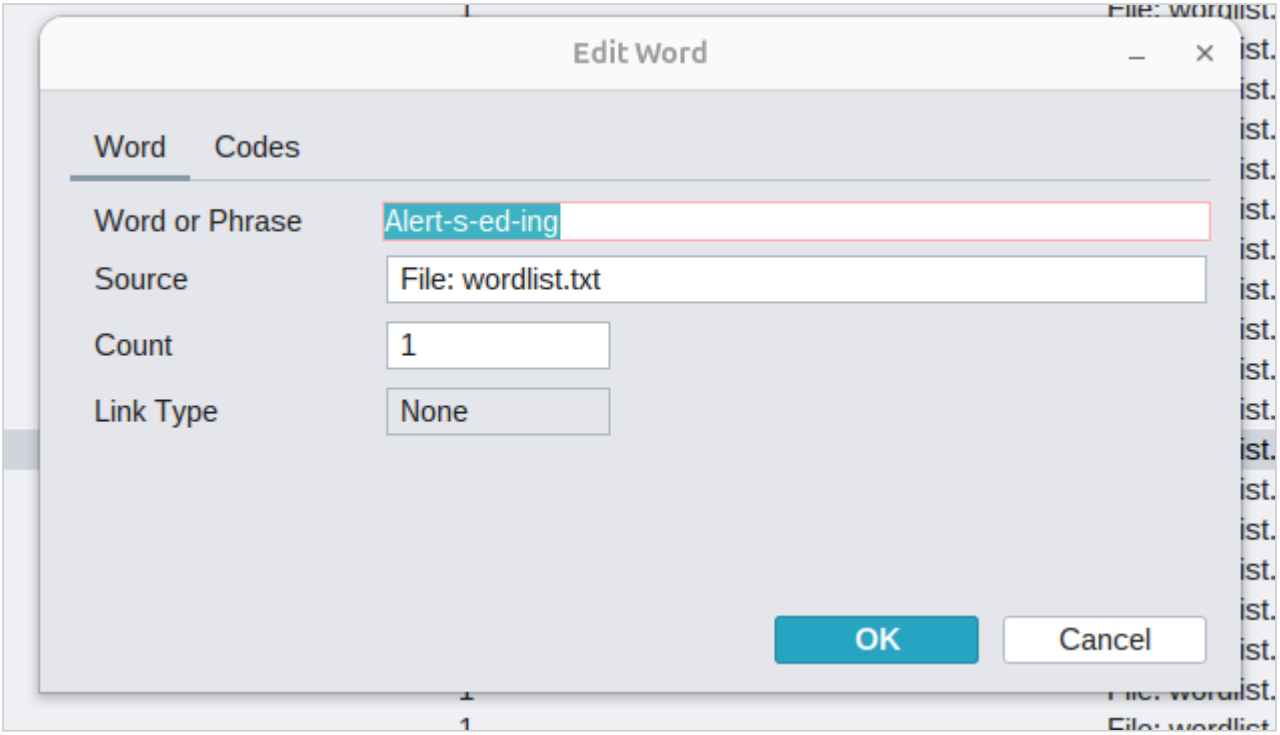
Actions

Button	Description
OK	Validates the form, saves changes, and closes the dialog. The Word or Phrase field is required.
Cancel	Discards changes and closes the dialog. Press `Escape` to cancel.

Related Topics

- Word List Panel
- Word List Settings

Word Edit Panel



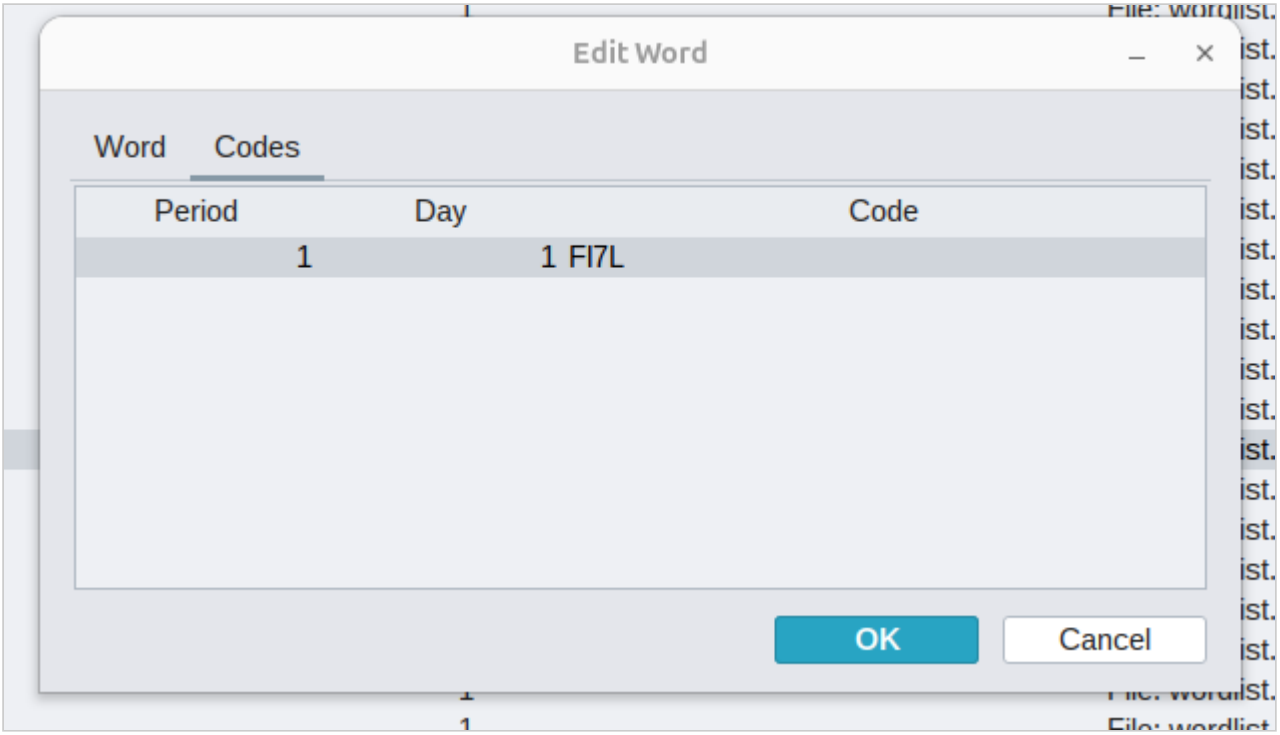
The Word tab in the Word Edit Dialog provides fields for editing word properties.

Fields

Field	Description
Word or Phrase	The word or phrase to include in the word list (required). This is the plaintext entry that will be assigned a generated code.
Source	The source or origin of this word entry (e.g., "Manual", "Frequency", "Location").
Count	The number of codes to generate for this word (1–99). Overrides the default from Word List Settings.
Link Type	Shows how this word is linked to other entities (read-only). Words linked from frequencies, locations, or documents cannot have their word text edited here — they are managed through their source entity.

> The Word or Phrase field is required. Linked words (those originating from a frequency, location, or document with "Add to Word List" enabled) display in bold and cannot be renamed from this dialog.

Word Codes



The Codes tab in the Word Edit Dialog displays the generated codes that have been assigned to this word. Codes are generated from the Word List Panel using the settings in Word List Settings.

This tab provides a read-only view of the codes. The number and format of codes depend on the generation mode, pattern, and count configured in the word list settings.

Related Topics

- Generation Modes
- Code Patterns
- Word List Settings

Word Notes

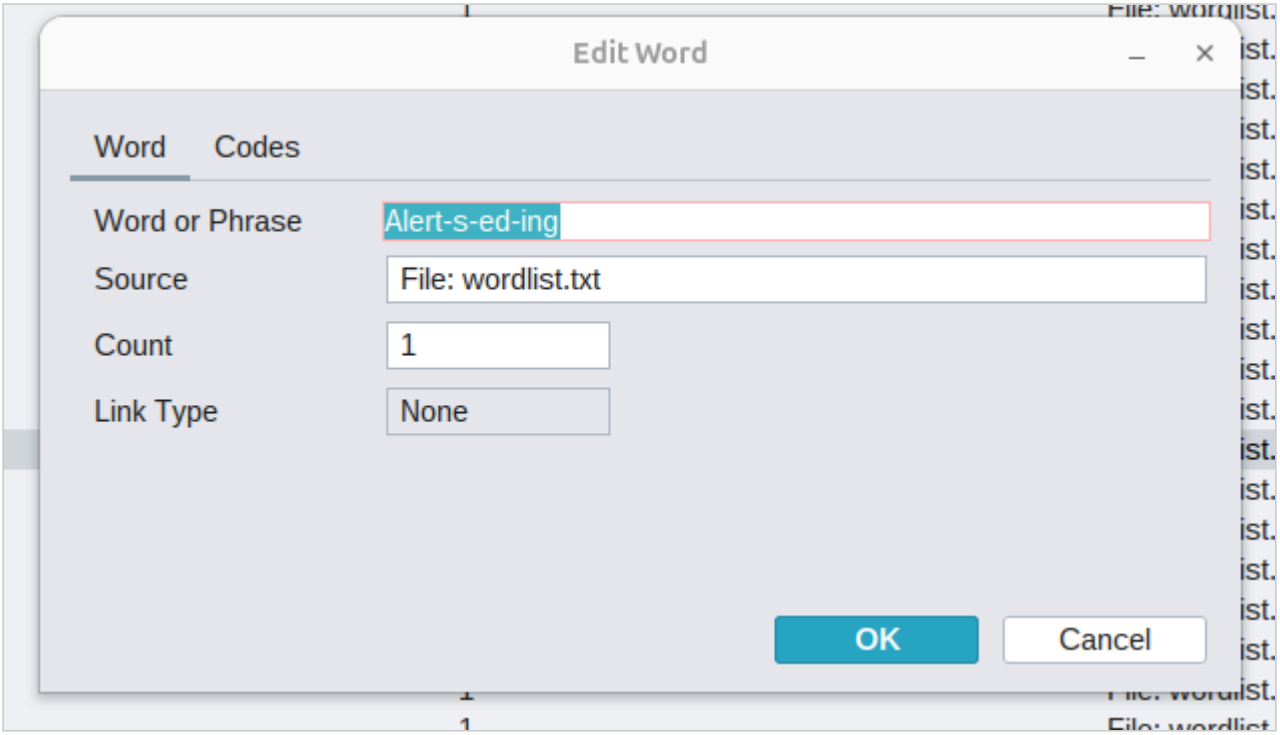
The Notes tab in the Word Edit Dialog provides a free-text area where you can record notes about the word list entry. Notes are saved with the word and included in the CEOI file.

> Notes are for the person building the CEOI. They are not printed on the Wordlist reports and never travel with a code.

Related Topics

- Word Edit Dialog
- Word Edit Tab
- Word Codes Tab
- Word List Panel

Word Edit Panel



The Word Edit Panel is the editing component used within the Word Edit Dialog. It provides fields for entering and modifying a word's properties including the word or phrase text, source, code count, and link type. See Word Edit Panel (Dialog) for a full description of each field.

Word List Message Encoder / Decoder



The Word List Message Encoder/Decoder (CODEC) lets you encode plain-text messages into word-list code and decode word-list-encoded messages back into plain text. Select the period and division to use the correct word list for the encoding period, then type or paste text into the input area and click **Encode** or **Decode**.

Period and Division

Use the **Period** and **Division** dropdowns to select which CEOI period and division's word list to use. Encoding and decoding must use the same period and division, or the result will be incorrect.

Encoding a Message

1. Select the correct period and division.
2. Type or paste the plain text message into the upper input area.
3. Click **Encode**.
4. The word-encoded version of the message appears in the lower output area.

Decoding a Message

1. Select the correct period and division.
2. Type or paste the word-encoded message into the upper input area.
3. Click **Decode**.
4. The decoded plain text appears in the lower output area.

Cut / Copy / Paste

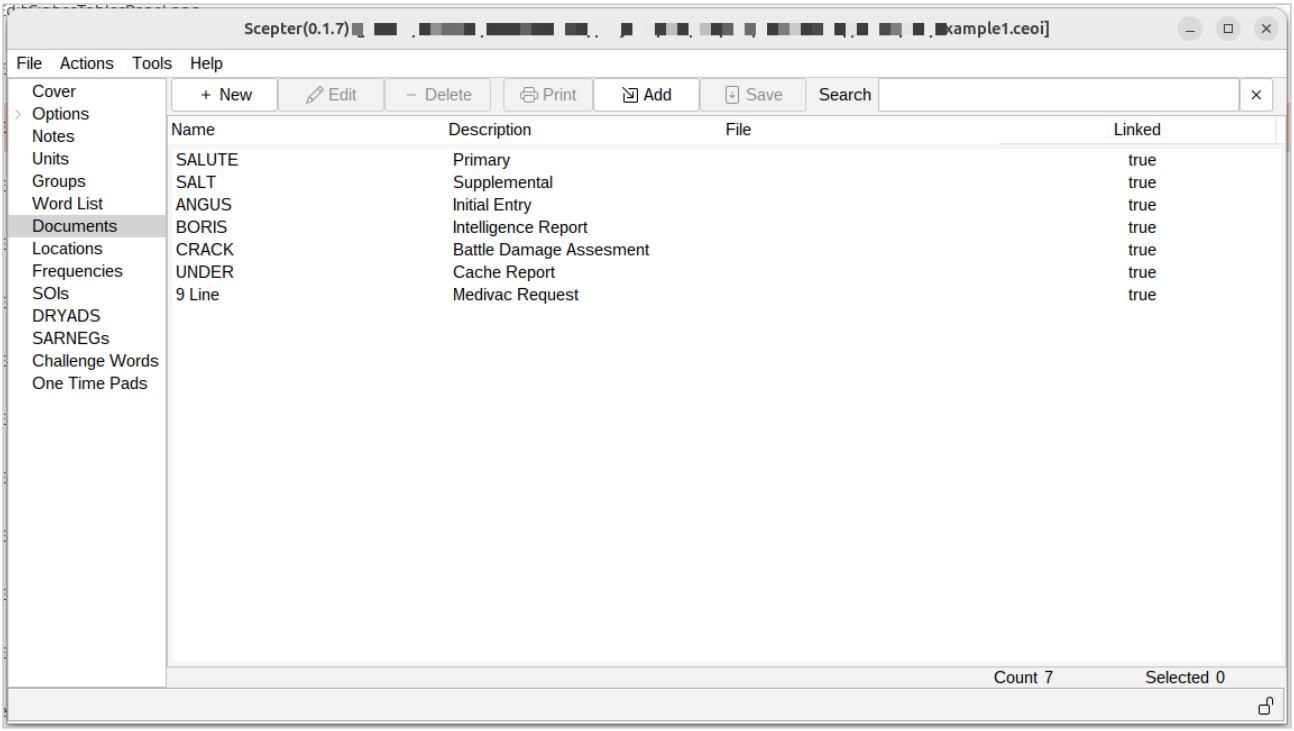
Right-click in the input area to access Cut, Copy, and Paste commands for the text. The output area is read-only but can be selected and copied to the clipboard.

> The word list changes each period and division. Always confirm the period and division match the intended recipients before encoding a message.

Related Topics

- CEOI Word List
- Word List Settings
- Word Edit Dialog

Documents



The Documents panel displays all document references defined in the current CEOI. Documents can represent SOPs, field manuals, overlays, or any other reference material that needs a code assignment for secure communication.

Table Columns

Column	Description
Name	The name or title of the document.
Description	A brief description of the document's contents or purpose.
File	The file path if a physical file is attached to this document entry.
Link	Indicates whether this document is also added to the word list for encoding.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new document. The Document Edit Dialog opens.
- **Add Files** – Click the **Add Files** button to quickly create document entries by selecting one or more files from disk. Each selected file becomes a new document entry with the filename as its name.
- **Edit** – Select a document and click the pencil button, press `Enter`, or double-click to edit it.
- **Delete** – Select one or more documents and click the **−** button or press `Delete`. If a document is linked to the word list, the corresponding word entry is also removed.

- **Save to Disk** – Select a document with an attached file and click the save button to export the embedded file to a folder on disk.

Search

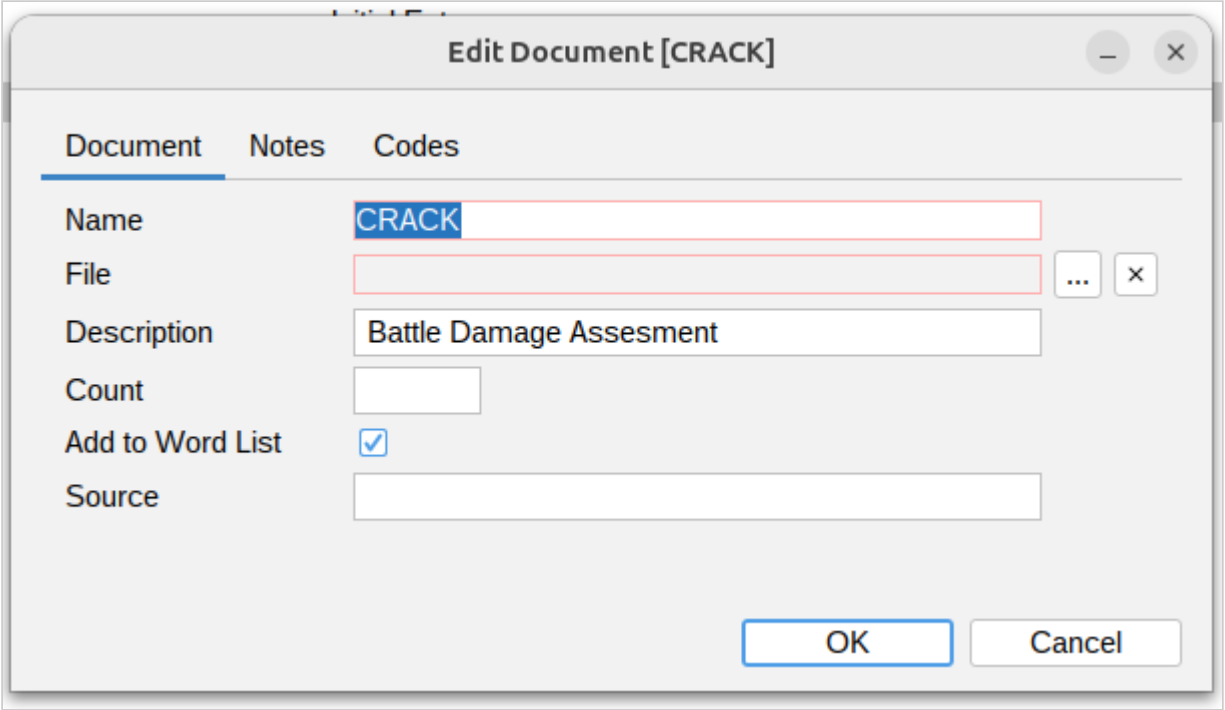
Type in the search field to filter documents by name. Clear the search to show all documents.

Import and Export

Use **Tools > Import List** and **Tools > Export List** (or the right-click context menu) to import or export document data.

> When a document has the "Add to Word List" option enabled, it automatically appears in the Word List so it can be referenced in encoded messages.

Document Edit Dialog



The Document Edit Dialog is used to create or edit a document entry in the CEOI. It is a tabbed dialog with tabs for editing document properties, notes, and viewing generated codes.

Tabs

Tab	Description
Document	Edit the document's name, file, description, and other properties. See Document Edit Panel.
Notes	Free-text notes about this document. See Document Notes.
Codes	View generated codes assigned to this document. See Document Codes.

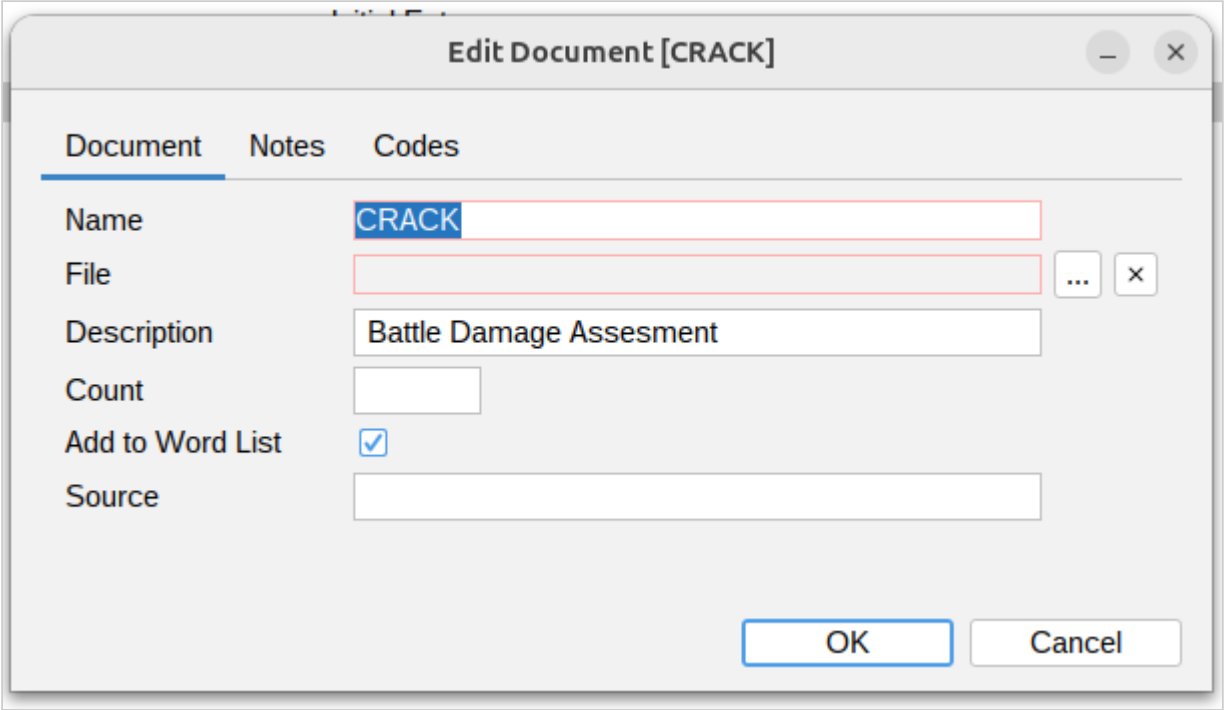
Actions

Button	Description
OK	Validates the form, saves changes, and closes the dialog. The Name field is required.
Cancel	Discards changes and closes the dialog. Press `Escape` to cancel.

Related Topics

- Documents Panel
- Documents Settings

Document Edit Panel



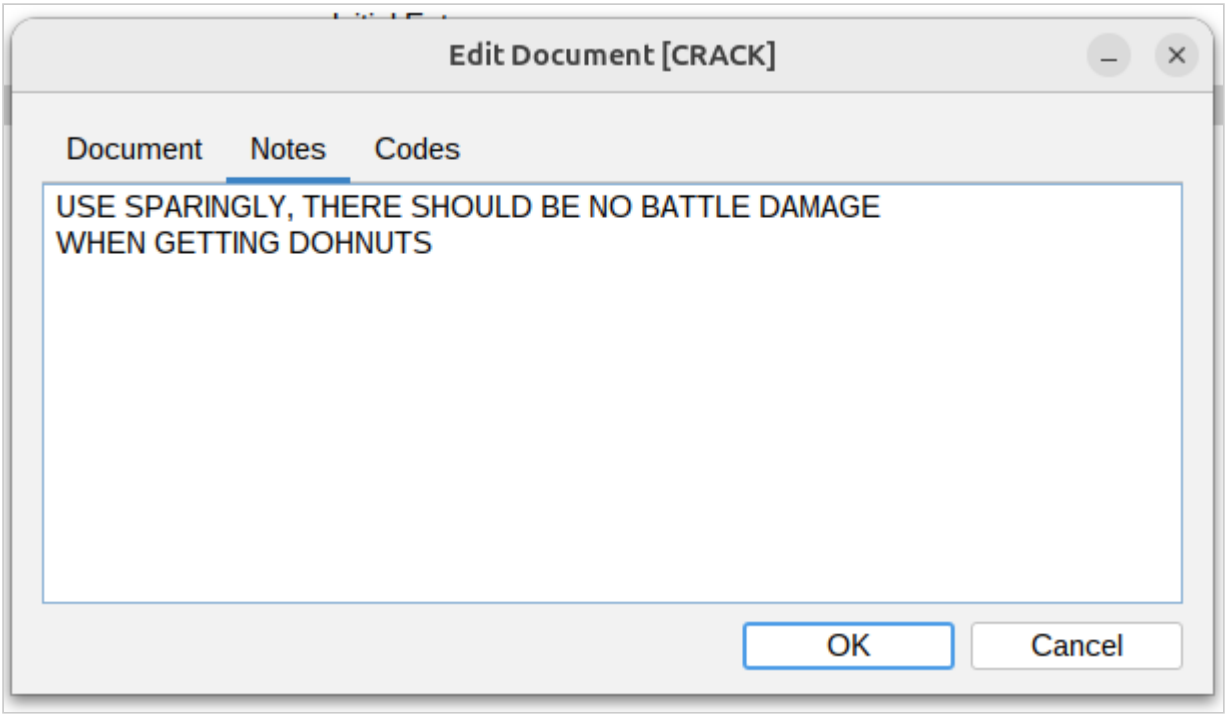
The Document tab in the Document Edit Dialog provides fields for editing document properties.

Fields

Field	Description
Name	The name or title of the document (required). This is shown in the documents list and reports.
File	The file path of the attached document (read-only). Click **...** to browse for a file, or click the **X** button to clear the attachment.
Description	A brief description of the document's contents or purpose.
Count	The number of codes to generate for this document (1–99). Overrides the default from Documents Settings.
Add to Word List	When checked, generated codes for this document are also added to the CEOI word list.
Source	The source or authority for this document.

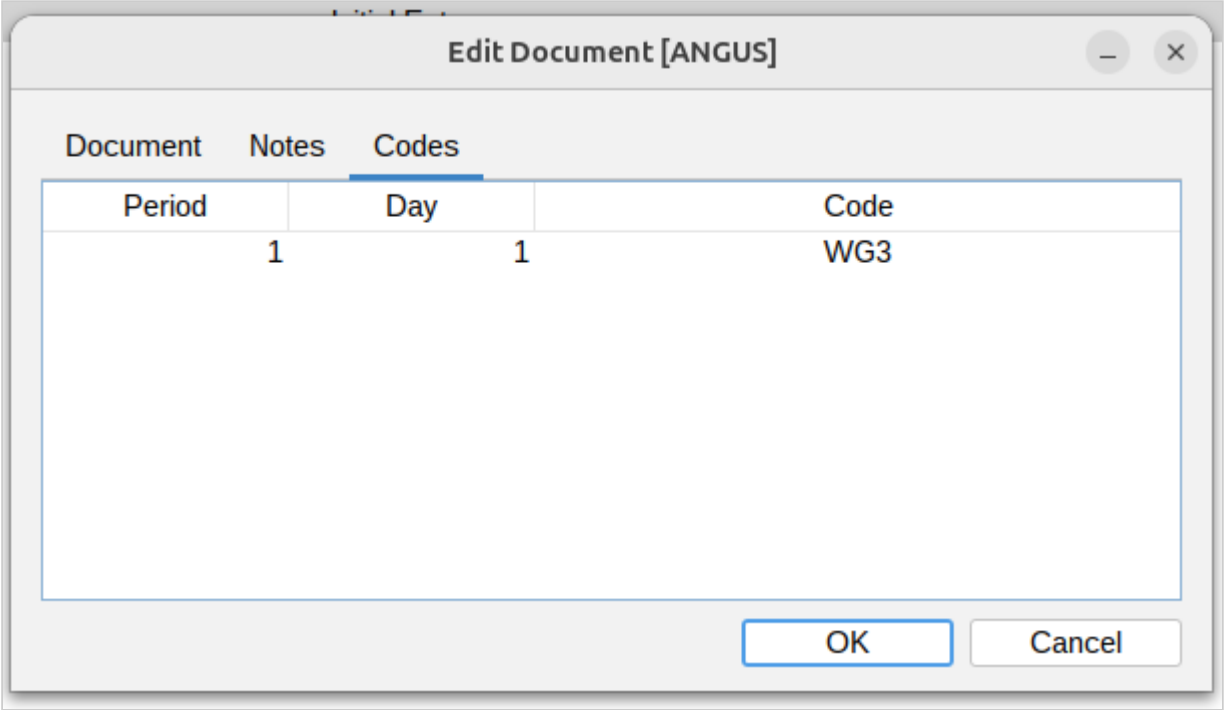
> The Name field is required. If left blank, a validation error will be displayed when you click OK.

Document Notes



The Notes tab in the Document Edit Dialog provides a free-text area where you can record notes about the document. Notes are saved with the document entity and included in the CEOI file.

Document Codes



The Codes tab in the Document Edit Dialog displays the codes assigned to this document. Codes are generated from the Documents Panel using the settings in Documents Settings.

The number and format of codes depend on the generation mode, pattern, and count configured in the documents settings.

Table Columns

Column	Description
Period	The CEOI period number for which this code is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Code	The assigned code. This column is editable: double-click a cell to type your own code. Codes you type are marked as overrides.
Override	Checked when the code was typed in rather than generated. It is set automatically when you edit a code.

Overriding Codes

Double-click a code and type a replacement, the same way you can override a unit callsign. The code is converted to uppercase and does not have to match the code pattern. If another document (or another row of this document) already uses the code, a duplicate warning is shown and the change is rejected. The change is saved when you click **OK**; **Cancel** discards it.

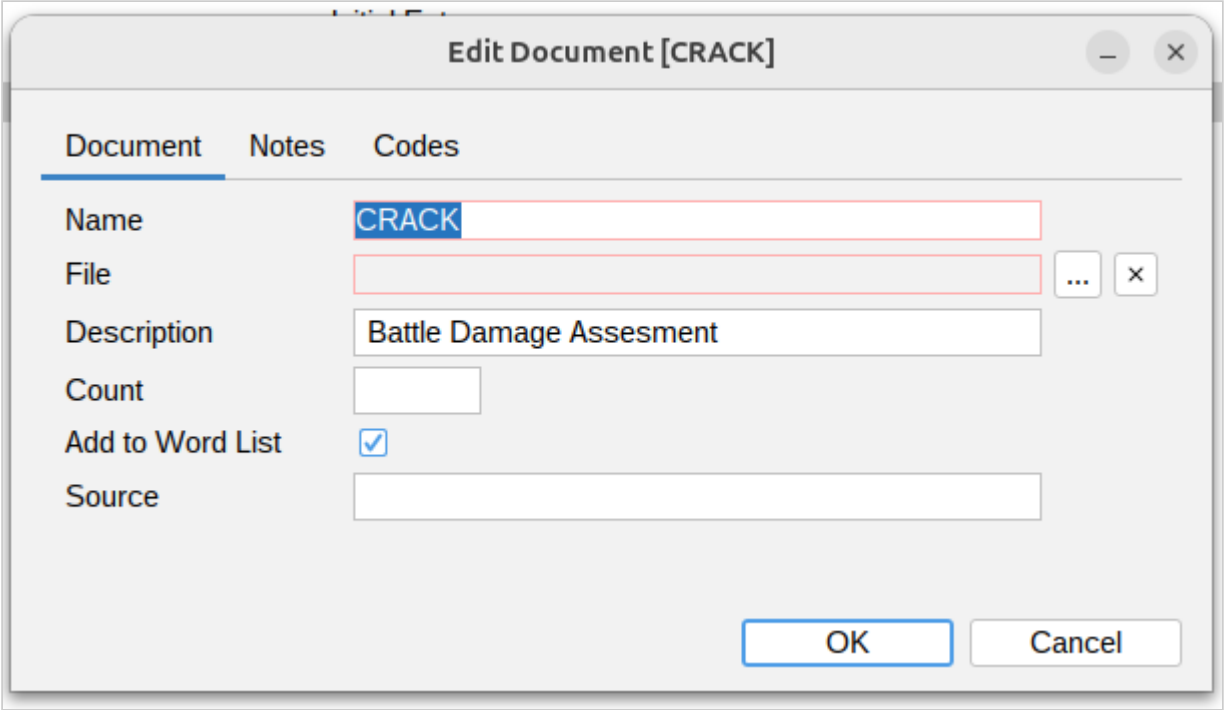
Overridden codes are kept when codes are generated again. If the code count is lowered, generated codes are removed first; overrides are removed only when there are more overrides than the new count allows. An override in a period or division the CEOI no longer has is removed.

> To remove overrides, use Clear CEOI Crypto and check **Document Code Overrides**. Checking **Documents** there clears only the generated codes.

Related Topics

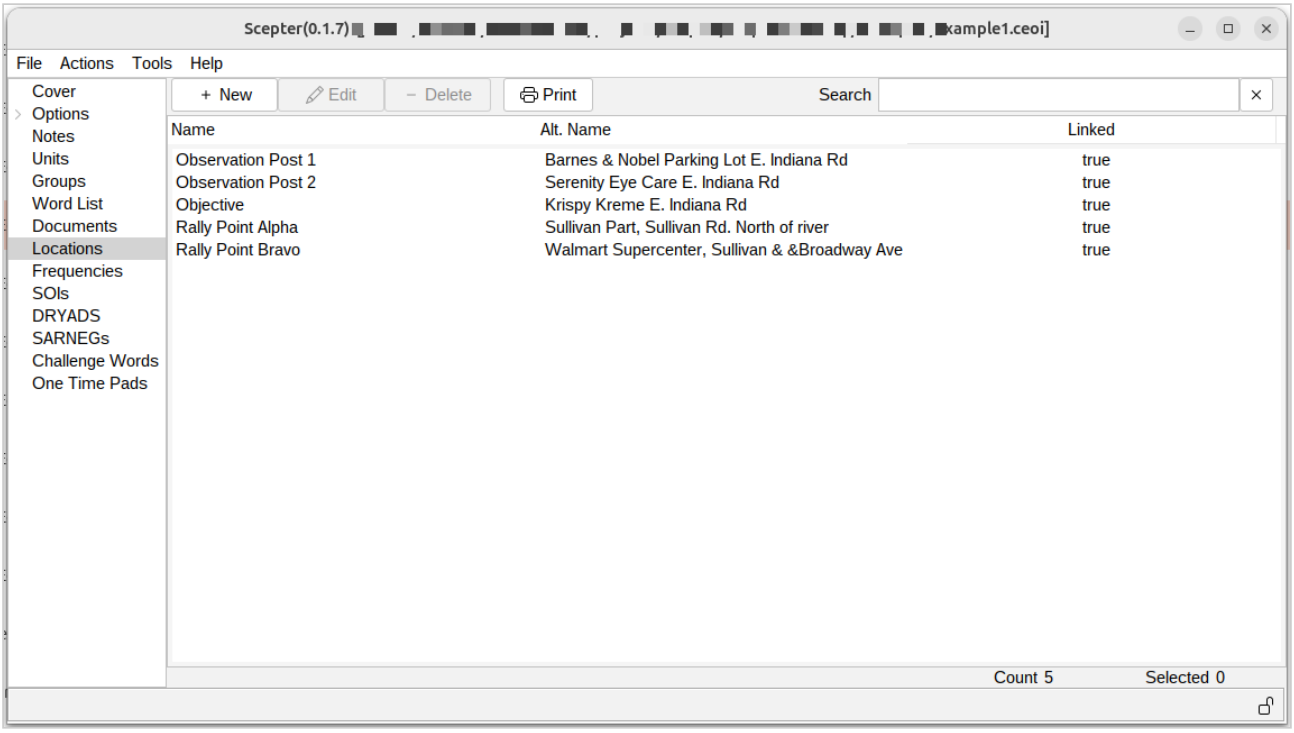
- Generation Modes
- Code Patterns
- Documents Settings

Document Edit Panel



The Document Edit Panel is the editing component used within the Document Edit Dialog. It provides fields for entering and modifying a document's properties including name, file attachment, description, source, code count, and word list inclusion. See Document Edit Panel (Dialog) for a full description of each field.

Locations



The Locations panel displays all named locations defined in the current CEOI. Locations can represent checkpoints, rally points, objectives, or any other geographic reference that needs a code assignment.

Table Columns

Column	Description
Name	The primary name of the location.
Alternate Name	An optional alternate or abbreviated name for the location.
Link	Indicates whether this location is also added to the word list for encoding.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new location. The Location Edit Dialog opens.
- **Edit** – Select a location and click the pencil button, press `Enter`, or double-click to edit it.
- **Delete** – Select one or more locations and click the **−** button or press `Delete`. If a location is linked to the word list, the corresponding word entry is also removed.

Search

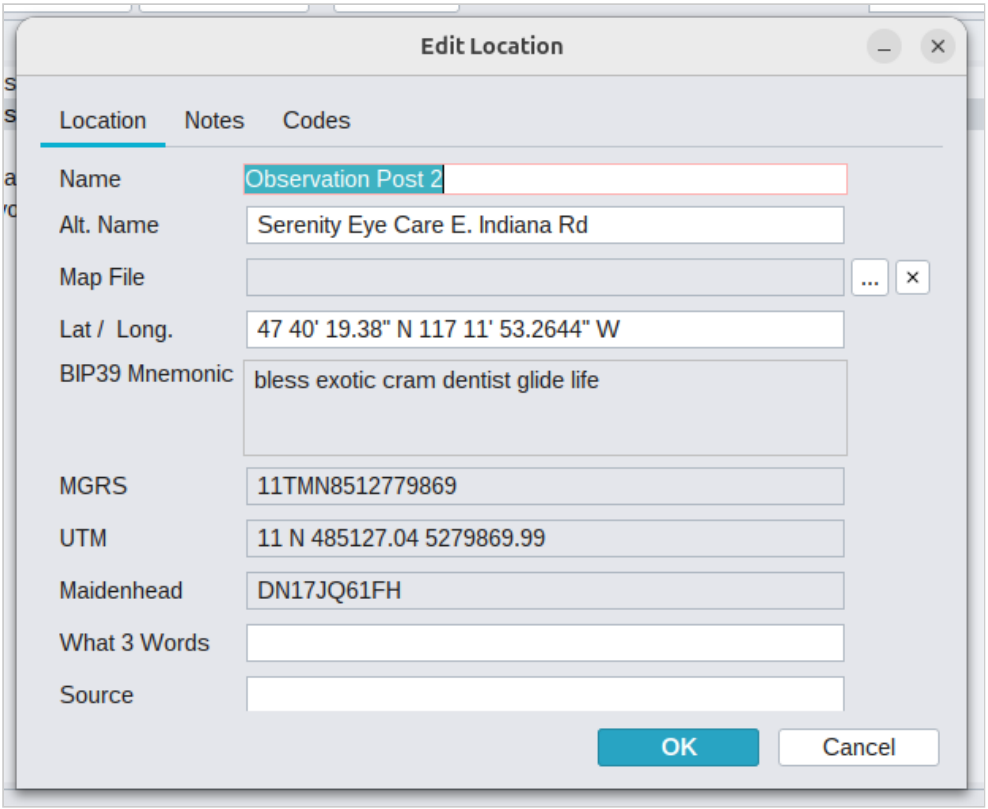
Type in the search field to filter locations by name. Clear the search to show all locations.

Import and Export

Use **Tools > Import List** and **Tools > Export List** (or the right-click context menu) to import or export location data.

> When a location has the "Add to Word List" option enabled, it automatically appears in the Word List so it can be referenced in encoded messages.

Location Edit Dialog



The Location Edit Dialog is used to create or edit a location entry in the CEOI. It is a tabbed dialog with tabs for editing location properties and coordinates, notes, and viewing generated codes.

Tabs

Tab	Description
Location	Edit the location's name, coordinates in multiple formats, and other properties. See Location Edit Panel.
Notes	Free-text notes about this location. See Location Notes.
Codes	View generated codes assigned to this location. See Location Codes.

Actions

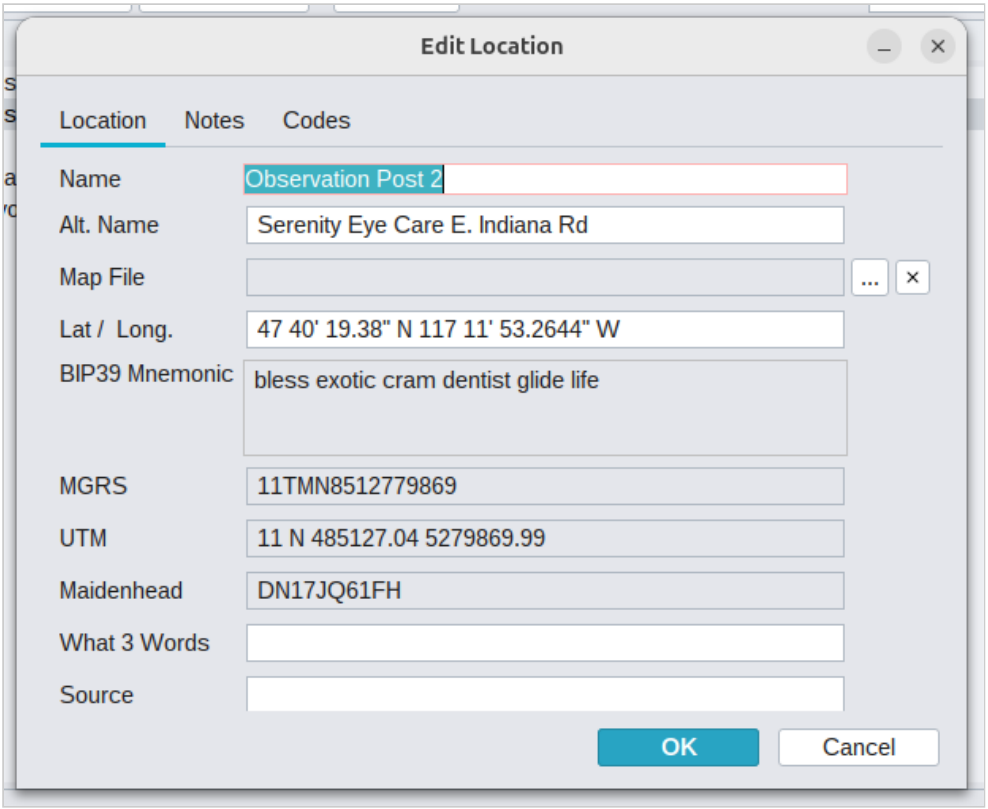
Button	Description
OK	Validates the form, saves changes, and closes the dialog. The Name field is required.
Cancel	Discards changes and closes the dialog. Press `Escape` to cancel.

Related Topics

- Locations Panel

- Locations Settings
- Coordinate Conversion Tool

Location Edit Panel



The Location tab in the Location Edit Dialog provides fields for editing location properties and coordinates in multiple formats.

Fields

Field	Description
Name	The name or designation for this location (required). This is shown in the locations list and reports.
Alt. Name	An alternate name for this location.
Map File	An optional map image file to attach to this location. Click **...** to browse for a file, or click the **X** button to clear the attachment.
Lat / Long.	Latitude and longitude coordinates in decimal degrees.
BIP39 Mnemonic	BIP 39 mnemonic word encoding of the location's coordinates.
MGRS	Military Grid Reference System coordinates.
UTM	Universal Transverse Mercator coordinates.
Maidenhead	Maidenhead Locator System grid square.
What 3 Words	What3Words geocoding address for this location.

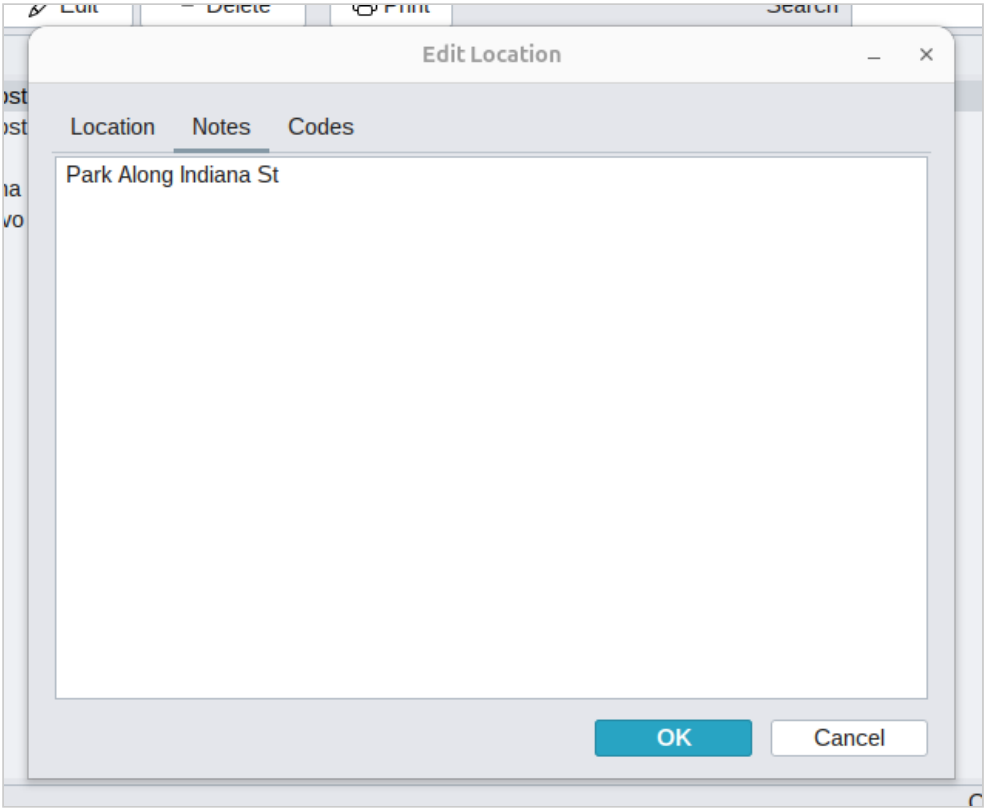
Source	The source or authority for this location data.
Count	The number of codes to generate for this location (1–99). Overrides the default from Locations Settings.
Add to Word List	When checked, generated codes for this location are also added to the CEOI word list.

Coordinate Entry

Which coordinate field is editable depends on the **Location Coordinate Preference** set in Locations Settings. When a preference is set, you enter coordinates in the preferred format and the other formats are automatically calculated. If the preference is set to "No Preference", all coordinate fields are editable.

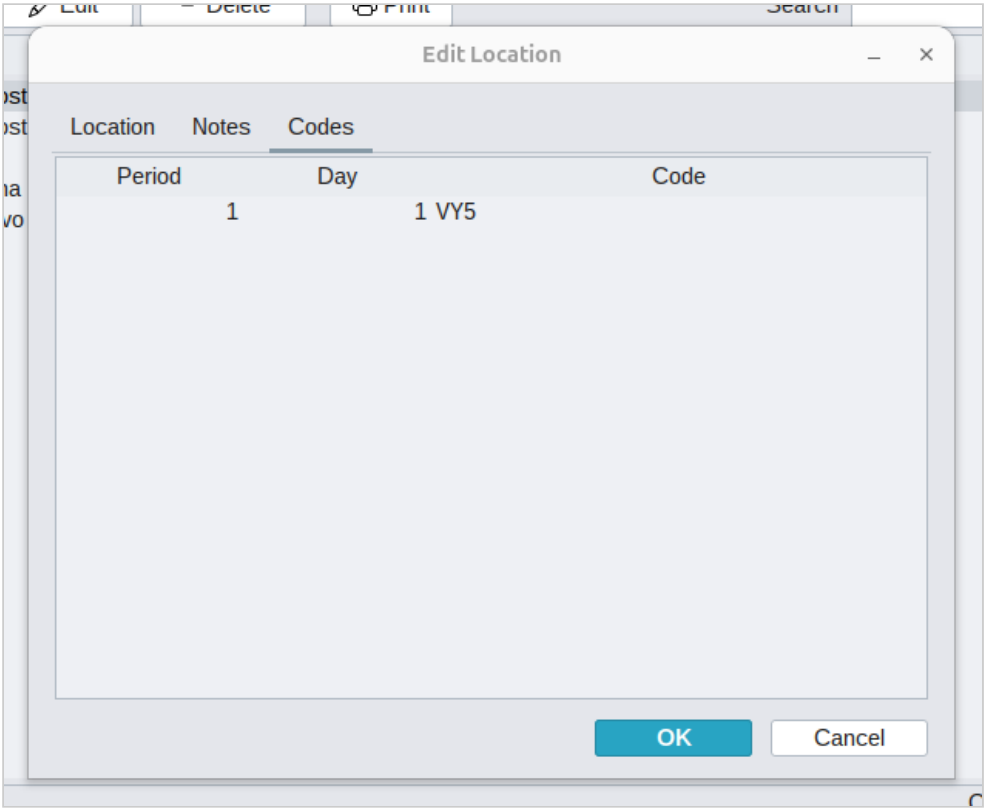
> The Name field is required. If left blank, a validation error will be displayed when you click OK.

Location Notes



The Notes tab in the Location Edit Dialog provides a free-text area where you can record notes about the location. Notes are saved with the location entity and included in the CEOI file.

Location Codes



The Codes tab in the Location Edit Dialog displays the codes assigned to this location. Codes are generated from the Locations Panel using the settings in Locations Settings.

The number and format of codes depend on the generation mode, pattern, and count configured in the locations settings.

Table Columns

Column	Description
Period	The CEOI period number for which this code is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Code	The assigned code. This column is editable: double-click a cell to type your own code. Codes you type are marked as overrides.
Override	Checked when the code was typed in rather than generated. It is set automatically when you edit a code.

Overriding Codes

Double-click a code and type a replacement, the same way you can override a unit callsign. The code is converted to uppercase and does not have to match the code pattern. If another location (or another

row of this location) already uses the code, a duplicate warning is shown and the change is rejected. The change is saved when you click **OK**; **Cancel** discards it.

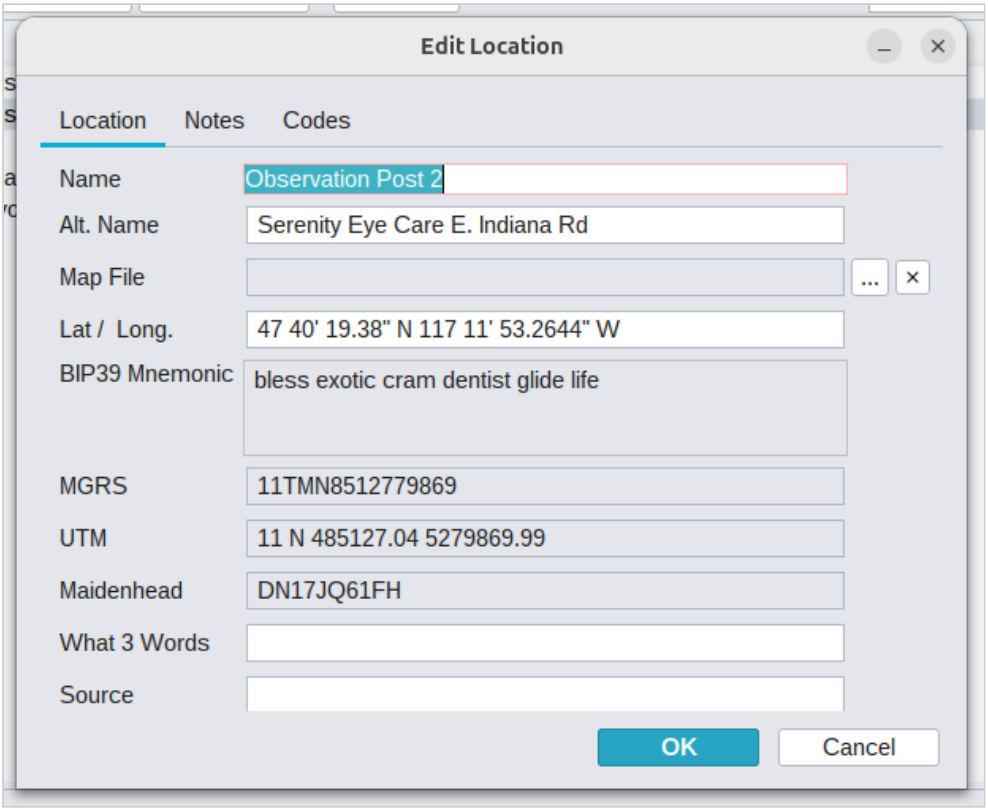
Overridden codes are kept when codes are generated again. If the code count is lowered, generated codes are removed first; overrides are removed only when there are more overrides than the new count allows. An override in a period or division the CEOI no longer has is removed.

> To remove overrides, use Clear CEOI Crypto and check **Location Code Overrides**. Checking **Locations** there clears only the generated codes.

Related Topics

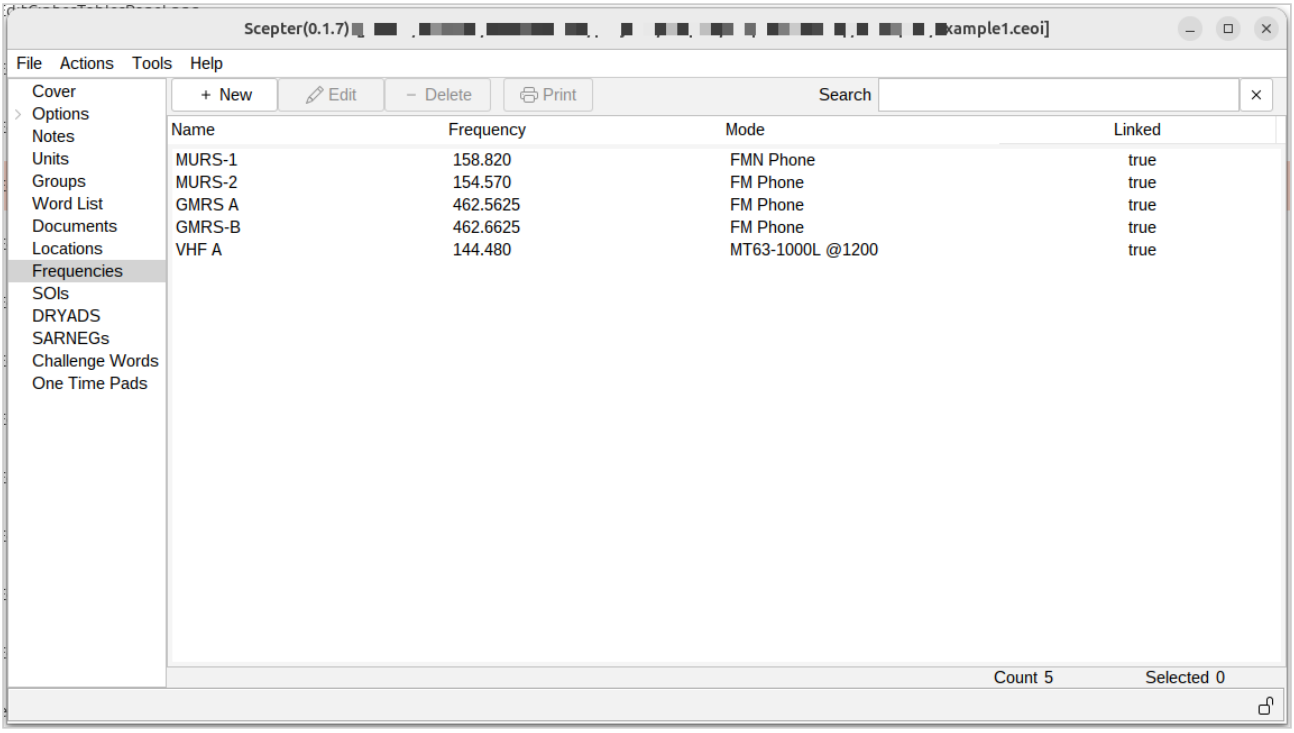
- Generation Modes
- Code Patterns
- Locations Settings

Location Edit Panel



The Location Edit Panel is the editing component used within the Location Edit Dialog. It provides fields for entering and modifying a location's properties including name, coordinates in multiple formats (Lat/Long, MGRS, UTM, Maidenhead, What3Words, BIP39), map file attachment, source, code count, and word list inclusion. See Location Edit Panel (Dialog) for a full description of each field.

Frequencies



The Frequencies panel displays all radio frequencies defined in the current CEOI. Frequencies can be assigned to units and groups as communication assets.

Table Columns

Column	Description
Name	The descriptive name for this frequency (e.g., "Company Command Net").
Frequency	The radio frequency value (e.g., "152.450 MHz").
Mode	The transmission mode (e.g., FM, AM, SSB).
Band	The band the frequency belongs to. When a frequency is created or edited it is put in the narrowest band whose range holds its frequency and whose mode matches.
Link	Indicates whether this frequency is also added to the word list for encoding.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new frequency. The Frequency Edit Dialog opens.
- **Edit** – Select a frequency and click the pencil button, press `Enter`, or double-click to edit it.
- **Delete** – Select one or more frequencies and click the **−** button or press `Delete`. If a frequency is linked to the word list, the corresponding word entry is also removed.

Search

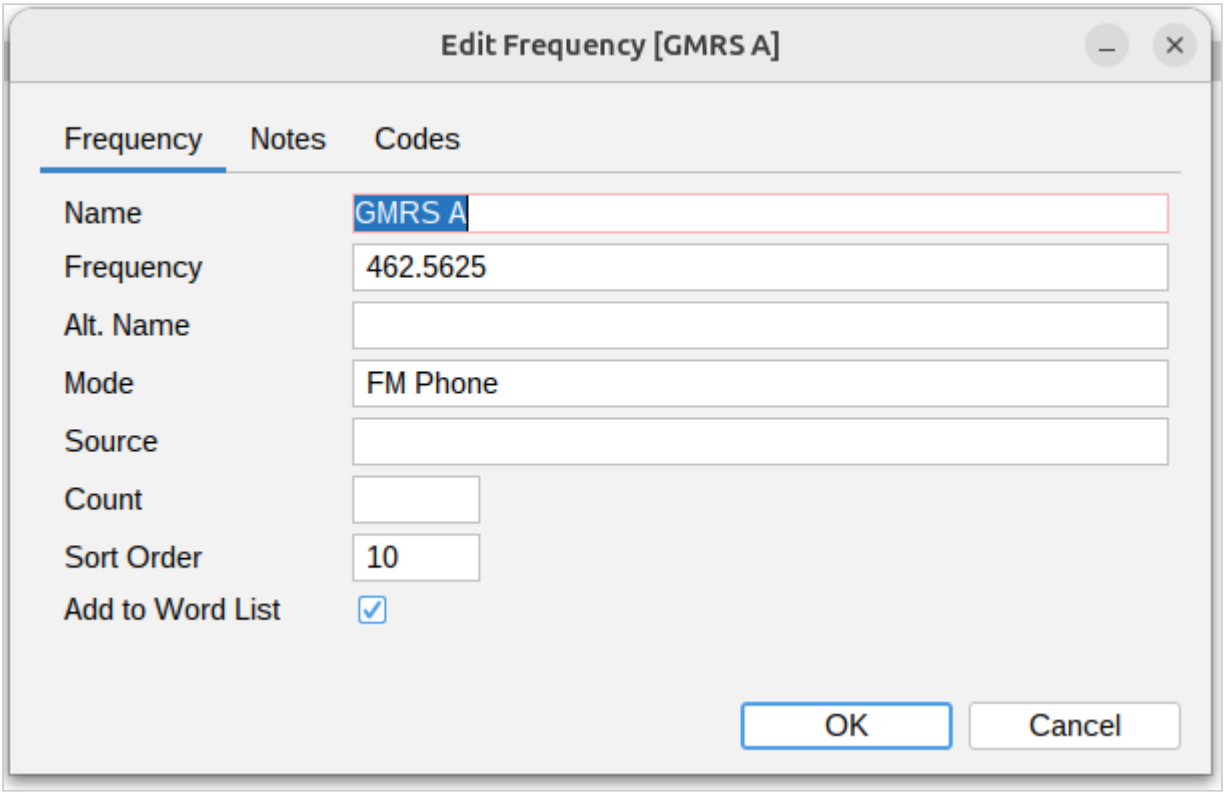
Type in the search field to filter frequencies by name. Clear the search to show all frequencies.

Import and Export

Use **Tools > Import List** and **Tools > Export List** (or the right-click context menu) to import or export frequency data.

> When a frequency has the "Add to Word List" option enabled, it automatically appears in the Word List so it can be referenced in encoded messages.

Frequency Edit Dialog



The Frequency Edit Dialog is used to create or edit a frequency entry in the CEOI. It is a tabbed dialog with tabs for editing frequency properties, notes, and viewing generated codes.

Tabs

Tab	Description
Frequency	Edit the frequency's name, frequency value, mode, and other properties. See Frequency Edit Panel.
Notes	Free-text notes about this frequency. See Frequency Notes.
Codes	View generated codes assigned to this frequency. See Frequency Codes.

Actions

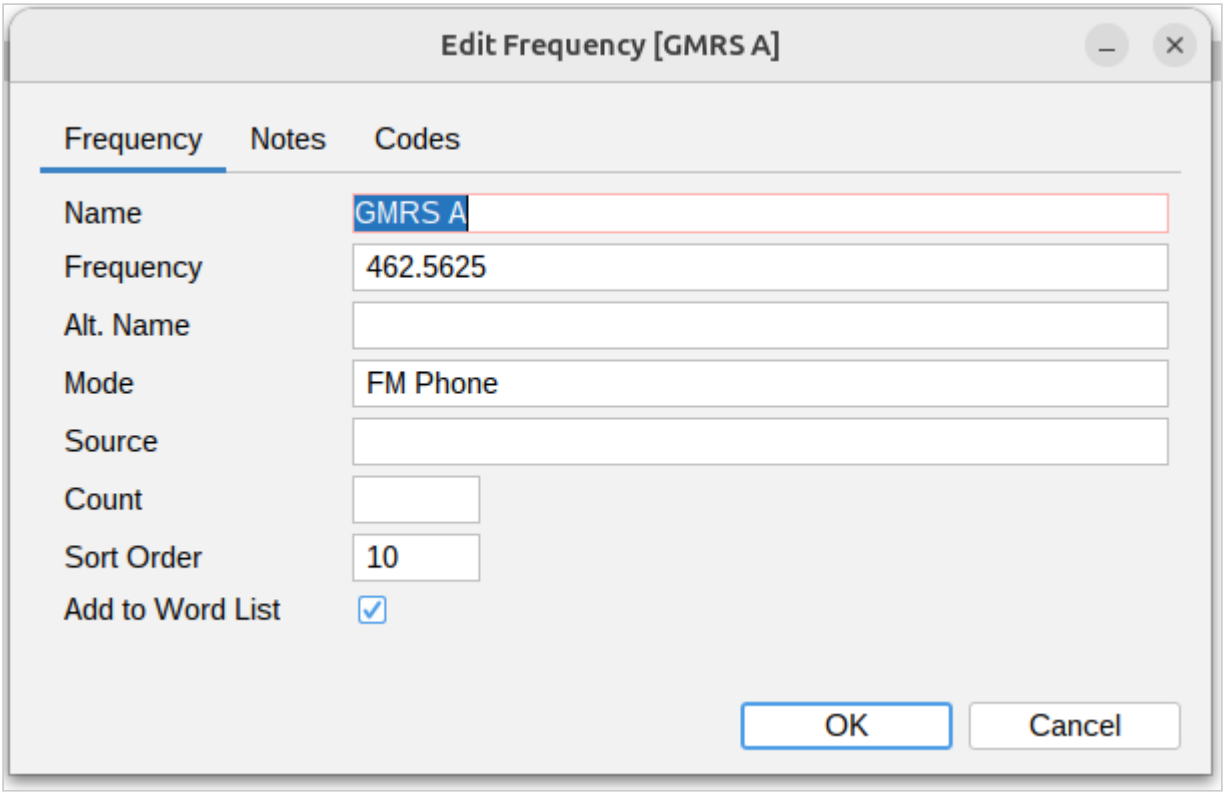
Button	Description
OK	Validates the form, saves changes, and closes the dialog. The Name field is required.
Cancel	Discards changes and closes the dialog. Press `Escape` to cancel.

Related Topics

- Frequencies Panel

- Frequencies Settings

Frequency Edit Panel



The Frequency tab in the Frequency Edit Dialog provides fields for editing frequency properties.

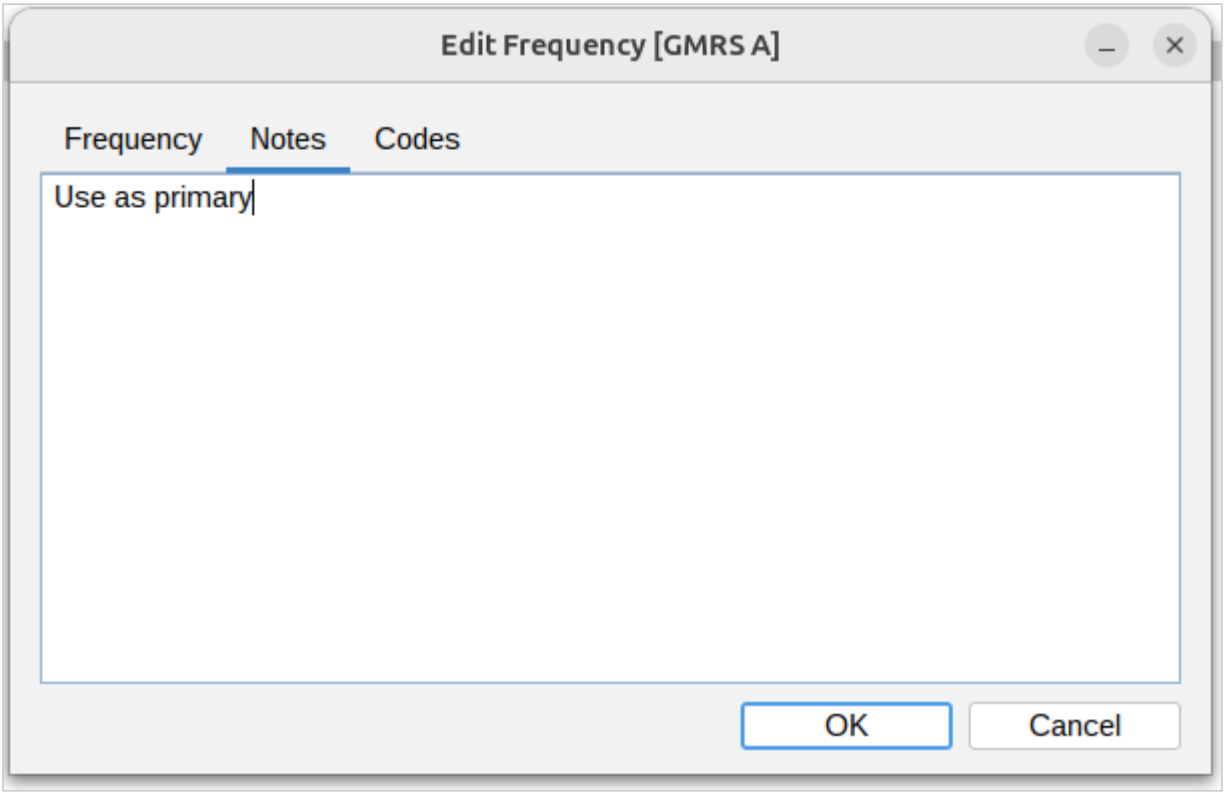
Fields

Field	Description
Name	The name or designation for this frequency (required). This is shown in the frequency list and reports.
Frequency	The radio frequency value (e.g., "146.520").
Band	Read only. The band the frequency is in. It is the narrowest band whose range holds the frequency and whose mode matches, and is filled in again when you leave the Frequency or Mode box, so you can see the band before you save.
Alt. Name	An alternate name or designation for this frequency.
Mode	The transmission mode (e.g., FM, AM, SSB, Digital).
Source	The source or authority for this frequency assignment.
Count	The number of codes to generate for this frequency (1–99). Overrides the default from Frequencies Settings.
Sort Order	Controls the display order of this frequency in lists and reports (1–9999).

Add to Word List	When checked, generated codes for this frequency are also added to the CEOI word list.

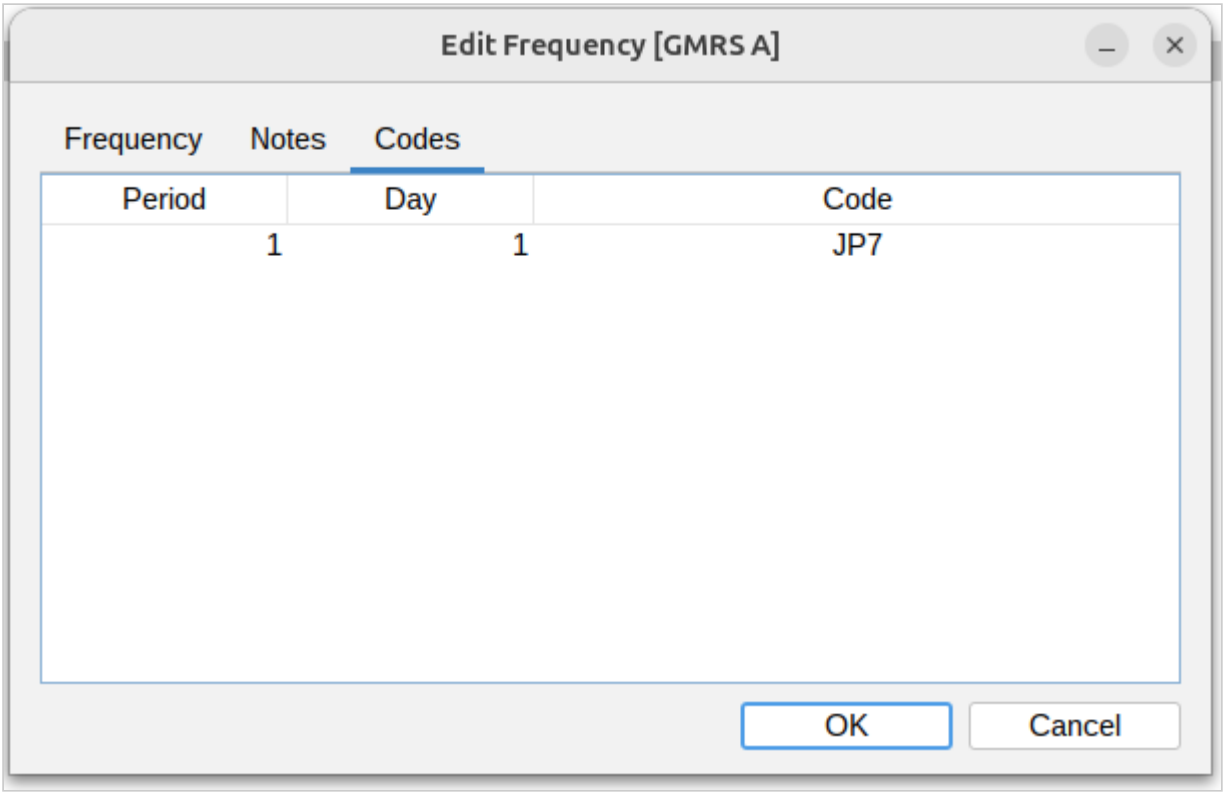
> The Name field is required. If left blank, a validation error will be displayed when you click OK.

Frequency Notes



The Notes tab in the Frequency Edit Dialog provides a free-text area where you can record notes about the frequency. Notes are saved with the frequency entity and included in the CEOI file.

Frequency Codes



The Codes tab in the Frequency Edit Dialog displays the codes assigned to this frequency. Codes are generated from the Frequencies Panel using the settings in Frequencies Settings.

The number and format of codes depend on the generation mode, pattern, and count configured in the frequencies settings.

Table Columns

Column	Description
Period	The CEOI period number for which this code is valid.
Division	The division (Day, Week, Month, etc.) within the period. The column header reflects the Division Units setting configured on the Cover panel.
Code	The assigned code. This column is editable: double-click a cell to type your own code. Codes you type are marked as overrides.
Override	Checked when the code was typed in rather than generated. It is set automatically when you edit a code.

Overriding Codes

Double-click a code and type a replacement, the same way you can override a unit callsign. The code is converted to uppercase and does not have to match the code pattern. If another frequency (or

another row of this frequency) already uses the code, a duplicate warning is shown and the change is rejected. The change is saved when you click **OK**; **Cancel** discards it.

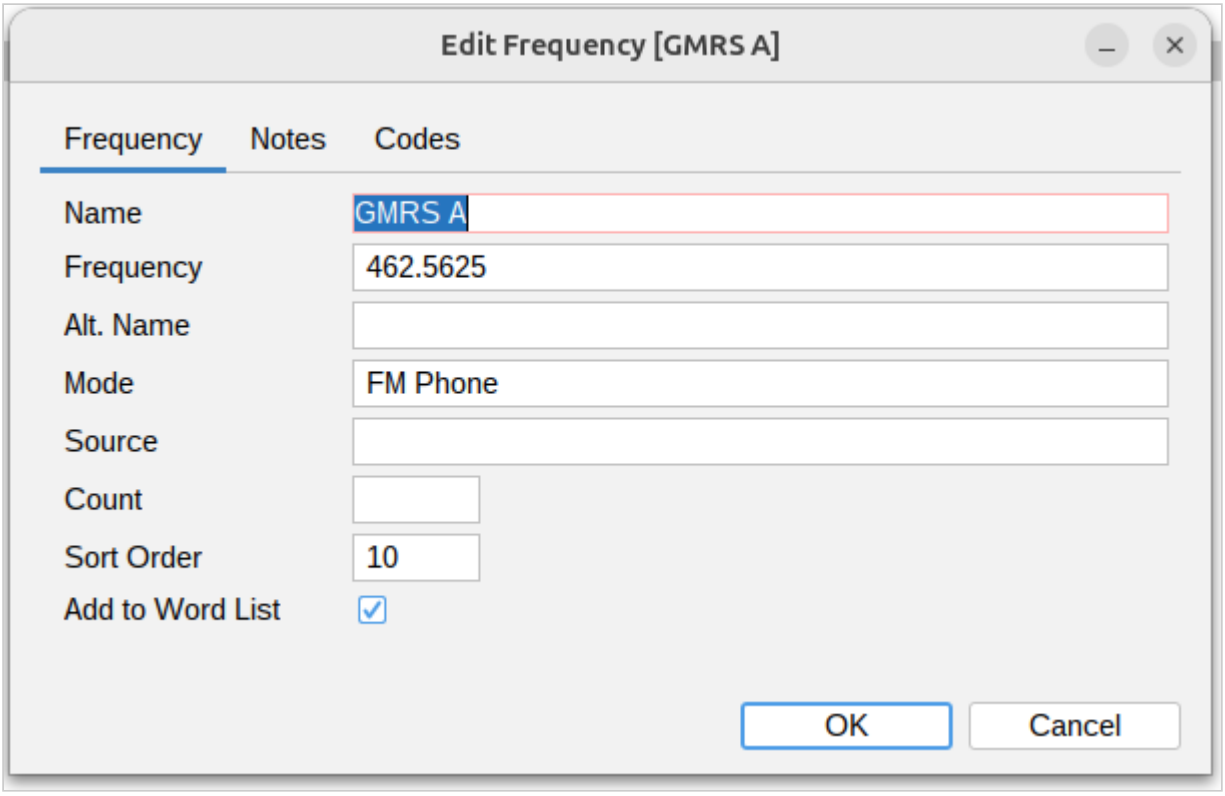
Overridden codes are kept when codes are generated again. If the code count is lowered, generated codes are removed first; overrides are removed only when there are more overrides than the new count allows. An override in a period or division the CEOI no longer has is removed.

> To remove overrides, use Clear CEOI Crypto and check **Frequency Code Overrides**. Checking **Frequencies** there clears only the generated codes.

Related Topics

- Generation Modes
- Code Patterns
- Frequencies Settings

Frequency Edit Panel



The Frequency Edit Panel is the editing component used within the Frequency Edit Dialog. It provides fields for entering and modifying a frequency's properties including name, frequency value, mode, source, code count, sort order, and word list inclusion. See Frequency Edit Panel (Dialog) for a full description of each field.

Bands

File > Settings > Bands... lists the frequency bands of the open CEOI. A band is a named range of frequencies used in one mode, such as *2m FM* from 144 to 148 in FM. Bands are saved with the CEOI and have no generated codes.

Table Columns

Column	Description
Name	The band's name, shown in the Band column of the Frequencies tab.
Start Frequency	The lowest frequency in the band.
End Frequency	The highest frequency in the band.
Mode	The mode a frequency must use to be in the band. Leave it blank for a band that takes any mode.
Description	Free text describing the band.

Actions

- **New** — opens the Band Editor on a new band.
- **Edit** — opens the selected band, when just one is selected. Double-clicking a row does the same.
- **Delete** — removes the selected bands. Select several with Ctrl-click, or a run of them with Shift-click; the Delete key does the same. Frequencies that were in a deleted band no longer belong to a band.
- **Import...** — adds the bands from a band plan file to the end of the list. See below.
- **Export...** — saves the selected bands to a band plan file, in the CSV format described below, so they can be imported into another CEOI. With no bands selected it saves a file holding just the header line, a blank band plan to fill in with a text editor or spreadsheet and then import. Scepter asks before replacing a file that already exists. A frequency typed with its unit, such as *146.52 MHz*, is saved as its number, and line breaks in a description become spaces.

Importing a Band Plan

Import... opens a file chooser on the band plans that ship with Scepter, in the `templates/bandplans` folder of the installation. Pick one, or a file of your own, and its bands are added after the ones already in the list. Nothing is replaced, so importing the same file twice lists its bands twice.

Scepter ships `arrl-us-amateur-band-plan.csv`, the US amateur radio band plan from the ARRL, with frequencies in MHz. It covers 160 m to 3 cm, with one band per segment and mode, using the modes CW, SSB, AM, FM, USB (60 m channels), DATA, SSTV and ATV. Each amateur band also has

an overall entry, such as *2 Meters VHF* from 144 to 148, that takes any mode, so a frequency that fits no segment still lands in its band.

These US channel plans also ship, each in its own file and in MHz. Each channel is a single frequency that takes any mode, and each file has overall entries that catch the frequencies between its channels:

- `us-cb-radio.csv` — the 40 Citizens Band channels and the whole 11 meter band.
- `us-marine-radio.csv` — the VHF marine channels, with a Ship Tx and a Coast Tx entry for each duplex channel, the NOAA weather channels, and the HF distress and calling frequencies.
- `us-commercial-business-radio.csv` — the business and industrial land mobile bands, and the itinerant business frequencies such as the color dot and color star frequencies.
- `us-general-use-radio.csv` — the FRS, GMRS and MURS channels, and the GMRS repeater inputs.

A band plan is a CSV file with one band per line:

```
Name,Start,End,Mode,Description
2m FM Calling,146.520,146.520,FM,National simplex calling frequency
2m FM Simplex,146.400,146.580,FM,Simplex
```

- **Name**, **Start** and **End** are required. Start and End are plain numbers, and Start must not be higher than End.
- **Mode** and **Description** may be left empty. A band with no mode takes any mode.
- A field containing a comma is put in double quotes.
- The header line is optional. Blank lines, and lines starting with `#`, are ignored.

The file is checked before anything is added. If any line is wrong, no bands are imported and Scepter lists the lines that need fixing.

How Frequencies Are Put in a Band

When a frequency is created, by **Add** or by importing a list, or edited, it is put in the narrowest band whose range holds its frequency (start and end included) and whose mode matches its own, ignoring case. So a calling frequency wins over the segment it sits in, and a segment wins over the whole band. Where two matching bands are the same width, one for the frequency's mode wins over one that takes any mode, and after that the one higher in the list. The order of the list does not otherwise matter. A frequency that matches no band is left without one. Frequencies are compared as numbers in whatever unit they were typed in, so enter bands in the same unit as your frequencies (for example MHz). Changing or adding a band does not move existing frequencies until they are next edited.

Related Topics

- Band Editor
- Frequencies

Band Editor

Creates or changes a band.

Fields

Field	Description
Name	Required. The name shown for the band.
Start Frequency	Required. The lowest frequency in the band, e.g. `144.000`.
End Frequency	Required. The highest frequency in the band, e.g. `148.000`. It must not be lower than the start frequency.
Mode	The mode a frequency must use to be in the band, e.g. `FM`. Matching ignores case. Leave it blank to accept any mode.
Description	Optional free text.

Use the same unit for the band as for your frequencies; the numbers are compared as typed.

When you create a band with **New**, the **Continuous Add** box appears beside the buttons. While it is ticked, the editor opens again on a blank band after each **OK**, so you can enter several bands in a row; **Cancel** stops. It starts out ticked or not according to **Continuous Add Mode** in the program settings. It is not shown when editing a band.

Related Topics

- Bands

Create Frequency List

Tools > Create Frequency List... adds frequencies to the Frequencies list, drawn at random from the bands you pick. The menu item is available once the CEOI has at least one band.

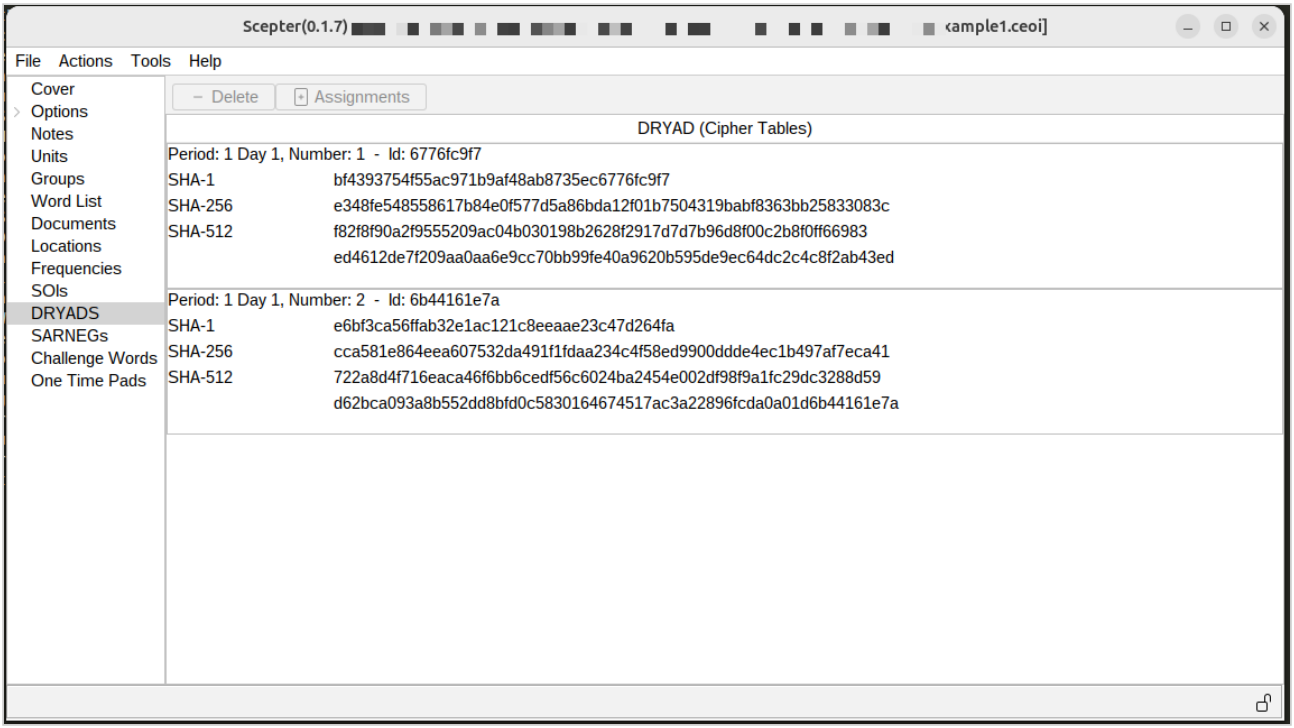
Fields

Field	Description
Bands	The bands to draw from. Select several with Ctrl-click, or a run of them with Shift-click.
Channel spacing	The gap between channels, in the same unit as the bands. For bands in MHz, 0.025 is 25 kHz and 0.0125 is 12.5 kHz. The channels start at the band's start frequency and go up to its end frequency.
Frequencies per band	How many frequencies to draw from each selected band.
Skip frequencies already in the frequency list	When ticked, a channel already used by a frequency in the list is not drawn.

What Is Created

- Each band's frequencies are picked at random from its channels, with no channel picked twice, even where selected bands overlap.
- A frequency is named after its band and numbered in frequency order, such as *2 Meters VHF 1*. A name already in the list is skipped.
- Its mode is the band's mode, blank for a band that takes any mode, and its source says which band it came from.
- Each frequency is then put in a band by the usual rule, the narrowest matching band, so it may land in a narrower band than the one it was drawn from.
- When a band has fewer free channels than asked for, all of them are used, and Scepter lists the bands that came up short.

DRYADs



The DRYADs panel displays all DRYAD tables in the current CEOI. DRYAD is a cipher system used to encrypt numerical information (such as grid coordinates) into letter groups for secure voice or written transmission.

DRYAD tables are generated automatically when you run **Actions > Generate Crypto**. The number and structure of generated tables depends on the settings in DRYAD Settings.

Display

Each DRYAD table is displayed as a compact grid in the table list. Double-click a table to open it in the DRYAD View Dialog for a full-size view.

Actions

- **Delete** – Select one or more tables and click the **−** button to remove them.
- **Assignments** – Select one or more tables and click the Assignments button to open the Asset Assignment Dialog, where you can assign DRYAD tables to specific units or groups.

Context Menu

Right-click a DRYAD table to access:

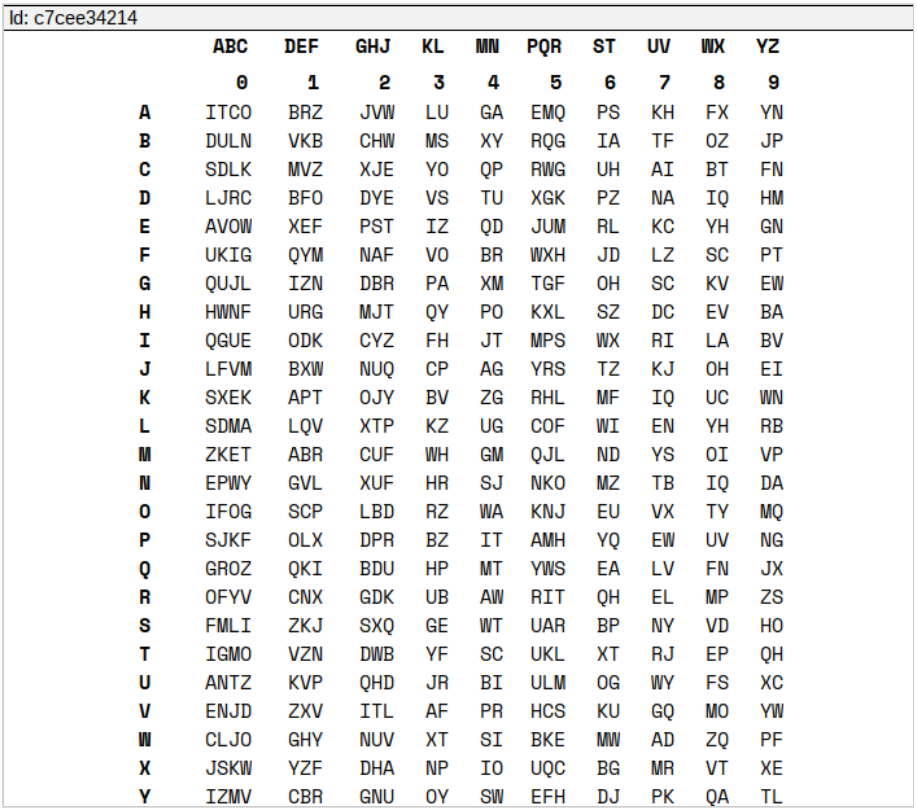
- **View** – Open the selected table in the DRYAD View Dialog.
- **Print** – Print the selected tables.
- **Assignments** – Manage unit/group assignments.

- **Copy Selected DRYADs to Clipboard** – Copy the selected table's text representation to the clipboard.
- **Copy SHA-1** – Copy the SHA-1 hash of the selected DRYAD to the clipboard.
- **Copy SHA-256** – Copy the SHA-256 hash of the selected DRYAD to the clipboard.
- **Copy SHA-512** – Copy the SHA-512 hash of the selected DRYAD to the clipboard.
- **Message Hash Tool...** – Opens the Message Hash Tool with this DRYAD pre-selected as the salt source.
- **Authenticate...** – Opens Authenticate with DRYAD on the first selected DRYAD.
- **Encrypt Numbers...** – Opens Encrypt Numbers with DRYAD on the first selected DRYAD.
- **Decrypt Numbers...** – Opens Decrypt Numbers with DRYAD on the first selected DRYAD.
- **Change Id...** – Prompts for a new identifier for the selected DRYAD and updates it immediately.
- **Delete Selected** – Remove the selected tables.

Related Topics

- DRYAD Viewer
- Message Hash Tool
- Asset Assignment Dialog
- DRYAD Settings

DRYAD Panel


Id: c7cee34214

	ABC	DEF	GHJ	KL	MN	PQR	ST	UV	WX	YZ
	0	1	2	3	4	5	6	7	8	9
A	ITCO	BRZ	JVW	LU	GA	EMQ	PS	KH	FX	YN
B	DULN	VKB	CHW	MS	XY	RQG	IA	TF	OZ	JP
C	SDLK	MVZ	XJE	YO	QP	RWG	UH	AI	BT	FN
D	LJRC	BFO	DYE	VS	TU	XGK	PZ	NA	IQ	HM
E	AVOW	XEF	PST	IZ	QD	JUM	RL	KC	YH	GN
F	UKIG	QYM	NAF	VO	BR	WXH	JD	LZ	SC	PT
G	QUJL	IZN	DBR	PA	XM	TGF	OH	SC	KV	EW
H	HWNF	URG	MJT	QY	PO	KXL	SZ	DC	EV	BA
I	QGUE	ODK	CYZ	FH	JT	MPS	WX	RI	LA	BV
J	LFVM	BXW	NUQ	CP	AG	YRS	TZ	KJ	OH	EI
K	SXEK	APT	OJY	BV	ZG	RHL	MF	IQ	UC	WN
L	SDMA	LQV	XTP	KZ	UG	COF	WI	EN	YH	RB
M	ZKET	ABR	CUF	WH	GM	QJL	ND	YS	OI	VP
N	EPWY	GVL	XUF	HR	SJ	NKO	MZ	TB	IQ	DA
O	IFOG	SCP	LBD	RZ	WA	KNJ	EU	VX	TY	MQ
P	SJKF	OLX	DPR	BZ	IT	AMH	YQ	EW	UV	NG
Q	GROZ	QKI	BDU	HP	MT	YWS	EA	LV	FN	JX
R	OFYV	CNX	GDK	UB	AW	RIT	QH	EL	MP	ZS
S	FMLI	ZKJ	SXQ	GE	WT	UAR	BP	NY	VD	HO
T	IGMO	VZN	DWB	YF	SC	UKL	XT	RJ	EP	QH
U	ANTZ	KVP	QHD	JR	BI	ULM	OG	WY	FS	XC
V	ENJD	ZXV	ITL	AF	PR	HCS	KU	GQ	MO	YW
W	CLJO	GHY	NUV	XT	SI	BKE	MW	AD	ZQ	PF
X	JSKW	YZF	DHA	NP	IO	UQC	BG	MR	VT	XE
Y	IZMV	CBR	GNU	OY	SW	EFH	DJ	PK	QA	TL

The DRYAD Panel displays a single DRYAD. It is used throughout Scepter wherever a DRYAD needs to be shown inline — for example inside the DRYAD viewer dialog and on the DRYADs settings panel.

Header Row

The header bar across the top of the panel shows the DRYAD's period, division, sequence number, and identifier (name). If the DRYAD has been assigned to any units or groups, a small assignment icon is displayed on the right side of the header.

Clicking the panel highlights it in blue to indicate it is selected.

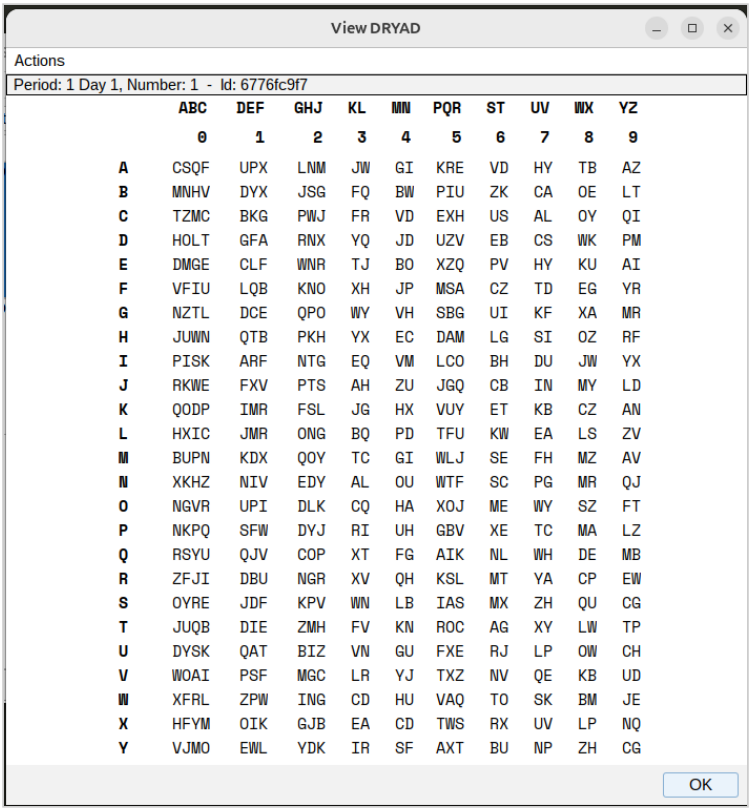
Table Rows

Below the header, 25 rows of cipher data are displayed. Each row contains the substitution mapping for that row of the DRYAD. The columns represent the cipher alphabet for encoding and decoding messages that reference this table.

Related Topics

- DRYAD Viewer
- DRYADs
- DRYAD Settings
- Asset Assignment Dialog

DRYAD Viewer



The DRYAD Viewer displays the full contents of a single DRYAD. It shows the header (period, division, number, and identifier) along with all 25 cipher rows. Use this dialog to inspect a DRYAD, copy it to the clipboard, generate a QR code, or manage its assignments.

Actions Menu

Action	Description
Assignments	Opens the Asset Assignment dialog to manage which units and groups are assigned to this DRYAD.
Copy to Clipboard	Copies the full text representation of the DRYAD (all print lines) to the system clipboard, ready to paste into any application.
Change ID	Prompts for a new identifier (name) for this DRYAD and updates it immediately.
QR Code	Generates a QR code containing the DRYAD's text plus its SHA1 hash. Opens the QR Code dialog.
Message Hash Tool...	Opens the Message Hash Tool with this DRYAD pre-selected as the salt source.
Authenticate...	Opens Authenticate with DRYAD on this DRYAD.
	Opens Encrypt Numbers with DRYAD on this DRYAD.

Encrypt Numbers...	
Decrypt Numbers...	Opens Decrypt Numbers with DRYAD on this DRYAD.

Right-Click Menu

The same actions are also available by right-clicking anywhere in the DRYAD area.

Closing the Viewer

Click **OK** or press **Escape** to close the viewer without making any changes.

> Use **Copy to Clipboard** to paste a DRYAD into a document or message for transmission. The SHA1 hash included in the QR Code option allows the recipient to verify the integrity of the table.

Related Topics

- DRYAD Panel
- DRYADs
- Asset Assignment Dialog
- QR Code Dialog
- Message Hash Tool

Authenticate with DRYAD

Tools > Authenticate with DRYAD... uses a DRYAD to prove that a station is who it says it is. It is also on the right-click menu of the DRYADs list and in the DRYAD Viewer's Actions menu, where it opens on that DRYAD. The menu item is available once the CEOI has DRYADs.

How a Challenge Works

The challenger sends two letters: a row letter from the left of the sheet, then a letter from that row, for example "authenticate ALFA JULIETT". The challenged station finds row A, finds J in it, and replies with the letter directly below it in row B. Only a station holding the same DRYAD can give the right reply.

Fields

Field	Description
DRYAD	The sheet in use, by period, division and number. Every station must use the same one.
Reply with	**the letter below**, the usual rule, or **the letter N places to the left** or **right** in the same row. Use whichever rule your stations agreed on. Below row Y is row A, right of a row's last letter is its first, and left of its first letter is its last. The rule applies to both tabs. The choice is remembered until Scepter closes.

Challenge Tab

For challenging another station. **New Challenge** picks two letters at random and shows them with their phonetic words, ready to send, together with the **Expected reply**. If the station answers with that letter, it is authenticated.

Reply Tab

For answering a challenge. Type the two letters you received, as letters (`AJ`) or phonetic words (`alfa juliett`). The reply to send appears as you type. If the row letter is not on the sheet, or the second letter is not in that row, Scepter says so instead.

> A station guessing a reply is right about one time in 25. Asking for two replies makes that one in 625, but uses the sheet twice as fast. Use a DRYAD only during its period.

Related Topics

- Encrypt Numbers with DRYAD
- Decrypt Numbers with DRYAD
- DRYAD Viewer
- DRYAD Settings

Encrypt Numbers with DRYAD

Tools > Encrypt Numbers with DRYAD... turns a short number, such as a grid reference, a frequency or a time, into letters using a DRYAD. It is also on the right-click menu of the DRYADs list and in the DRYAD Viewer's Actions menu.

How It Works

The two **indicator** letters are a row letter and a letter in that row. The letter just to the right of the second one is the **SET LETTER**, and its row encrypts the message. Each digit becomes one of the letters in that row's column for the digit. A digit that repeats gets a different letter each time, so repeats do not show in the ciphertext. The indicator is sent first, so the receiver can find the same row.

Fields

Field	Description
DRYAD	The sheet in use. The receiver must use the same one.
Indicator	Two letters picked at random when the dialog opens. **New** picks two more, or type your own. The SET LETTER they name is shown beside them.
Number	The number to encrypt. Points, slashes, dashes and spaces are kept as they are, so `146.520` stays in the same shape. Letters cannot be encrypted: take them out and send them separately.
Send	The indicator, then the ciphertext, for example `AJ KXQ.RTM`. **Copy** puts it on the clipboard, **QR Code** shows it as a QR code, and **Encrypt Again** picks the letters again.

> A column holds two to four letters. When a digit appears more often than its column has letters, Scepter has to use a letter twice and warns you. Splitting a long number into two messages, each with its own indicator, avoids this.

Related Topics

- Decrypt Numbers with DRYAD
- Authenticate with DRYAD
- DRYAD Settings

Decrypt Numbers with DRYAD

Tools > Decrypt Numbers with DRYAD... turns letters encrypted with a DRYAD back into the number. It is also on the right-click menu of the DRYADs list and in the DRYAD Viewer's Actions menu.

Fields

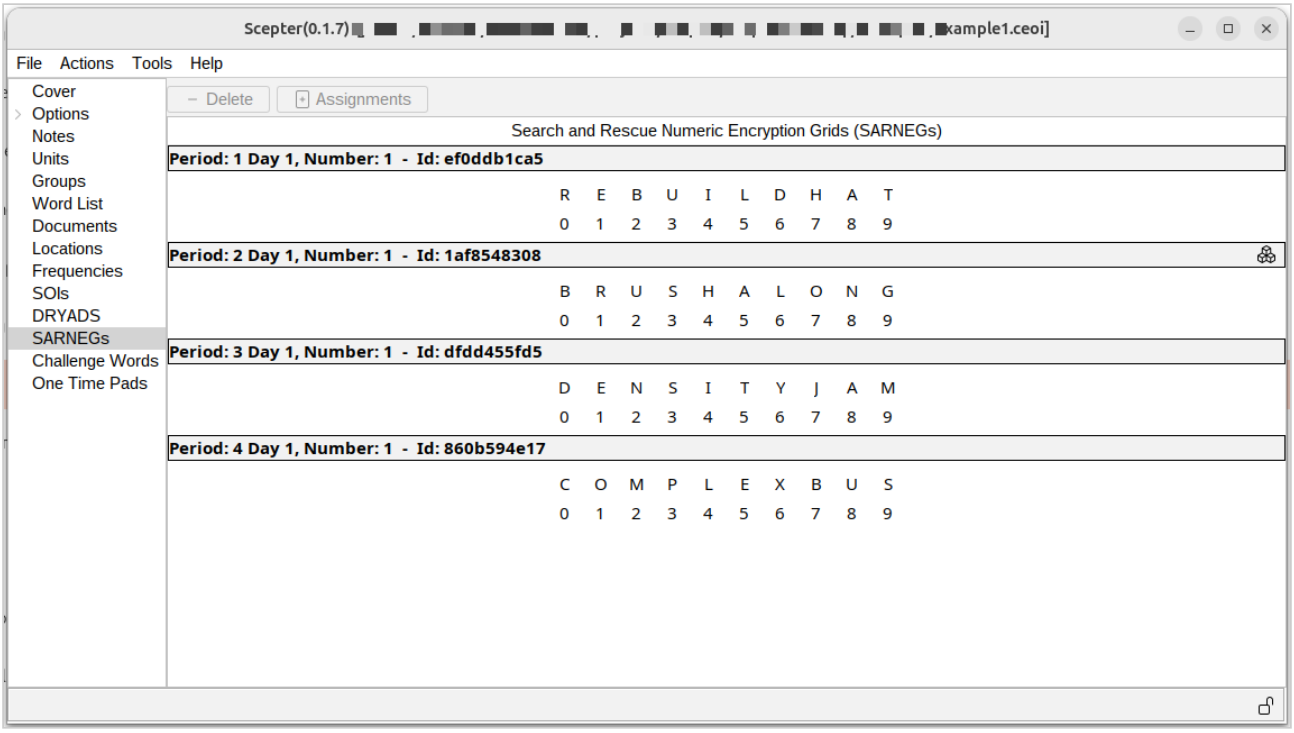
Field	Description
DRYAD	The sheet the sender used.
Indicator	The two letters sent ahead of the ciphertext. The SET LETTER they name is shown beside them.
Ciphertext	The letters received. With the Indicator field left empty, you can paste the whole message as Encrypt writes it, indicator first: `AJ KXQ.RTM`.
Number	The decrypted number, which updates as you type. **Copy** puts it on the clipboard.

Each letter is looked up in the SET LETTER's row, and the digit heading its column is the plain digit. A letter that is not in that row, usually a mistake in reception, is shown as `?`, and Scepter says how many there are.

Related Topics

- Encrypt Numbers with DRYAD
- Authenticate with DRYAD
- DRYAD Settings

SARNEGs



The SARNEGs panel displays all SARNEG (Search And Rescue Numerical Encryption Grid) tables in the current CEOI. SARNEGs are grid-based encryption tools used for encoding numerical data such as coordinates and grid references.

SARNEG tables are generated automatically when you run **Actions > Generate Crypto**. The number and structure of generated SARNEGs depends on the settings in SARNEG Settings.

Display

Each SARNEG is displayed as a compact grid cell in the table, showing its header and grid contents. Double-click a SARNEG to open it in the SARNEG View Dialog for a full-size view.

Actions

- **Delete** – Select one or more SARNEGs and click the **−** button to remove them.
- **Assignments** – Select one or more SARNEGs and click the Assignments button to open the Asset Assignment Dialog, where you can assign SARNEGs to specific units or groups.

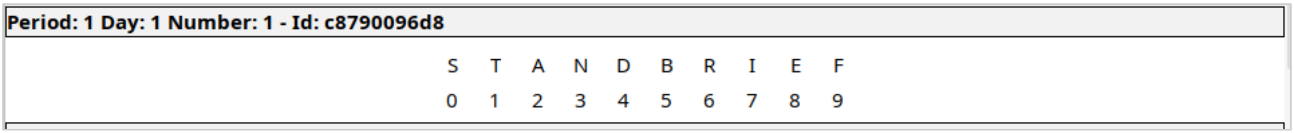
Context Menu

Right-click a SARNEG to access:

- **View** – Open the selected SARNEG in the SARNEG View Dialog.
- **Print** – Print the selected SARNEGs.
- **Assignments** – Manage unit/group assignments.
- **Change Id...** – Prompts for a new identifier for the selected SARNEG and updates it immediately.

• **Delete** – Remove the selected SARNEGs.

SARNEG Panel



The SARNEG Panel displays a single SARNEG (phonetic word-to-digit mapping). It is used throughout Scepter wherever a SARNEG needs to be shown inline — for example inside the SARNEG viewer dialog and on the SARNEG settings panel.

Header Row

When a header is shown, it displays the SARNEG's period, division, sequence number, and identifier (name). If the SARNEG has been assigned to any units or groups, an assignment icon appears on the right side of the header.

Letter-to-Digit Mapping

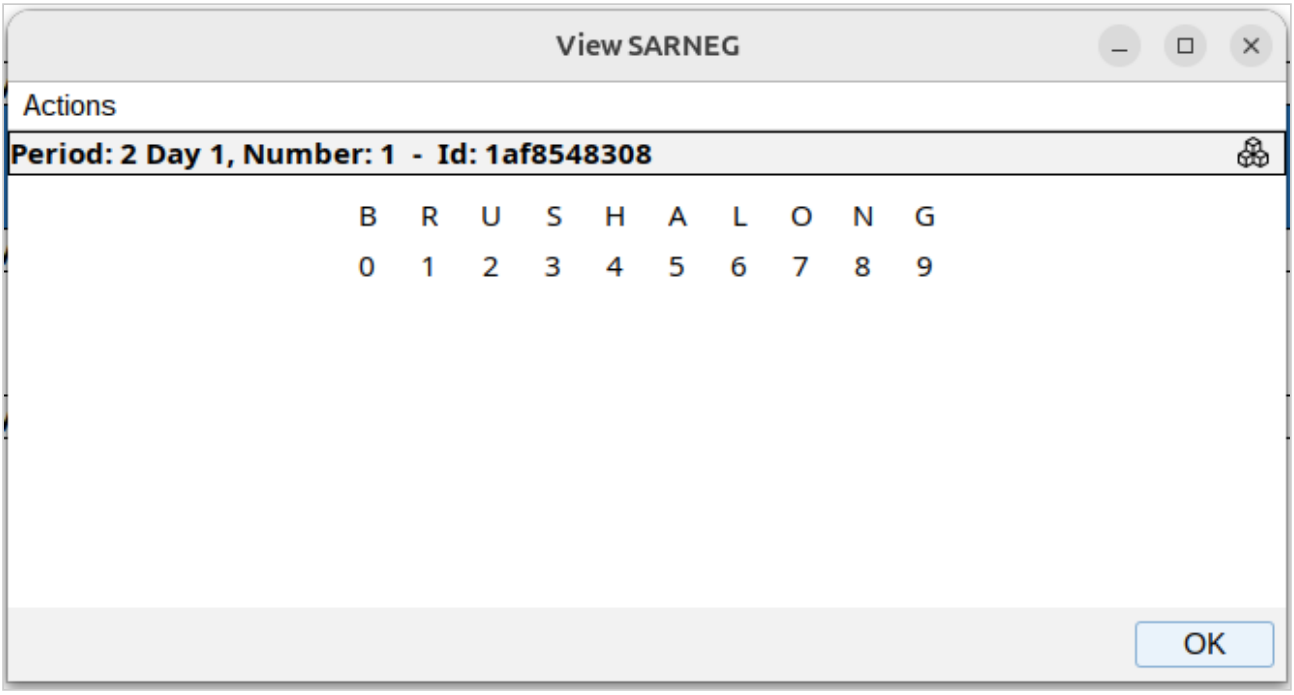
The body of the panel shows the phonetic word for each digit 0 through 9. The letters appear in the top row and the corresponding digit values (0–9) appear in the bottom row. This provides a compact view of the full SARNEG mapping.

Clicking the panel highlights it in blue to indicate it is selected.

Related Topics

- SARNEG Viewer
- CEOI SARNEGs
- SARNEG Settings
- Asset Assignment Dialog

SARNEG Viewer



The SARNEG Viewer displays the full contents of a single SARNEG phonetic code. It shows the header (period, division, number, and identifier) along with the complete letter-to-digit mapping. Use this dialog to inspect a SARNEG, copy it to the clipboard, or manage its assignments.

Actions Menu

Action	Description
Assignments	Opens the Asset Assignment dialog to manage which units and groups are assigned to this SARNEG.
Copy	Copies the SARNEG word and its digit mapping (e.g., WORD followed by 0123456789) to the system clipboard.
Change ID	Prompts for a new identifier (name) for this SARNEG and updates it immediately.

Right-Click Menu

The same actions are also available by right-clicking anywhere in the SARNEG panel area.

Closing the Viewer

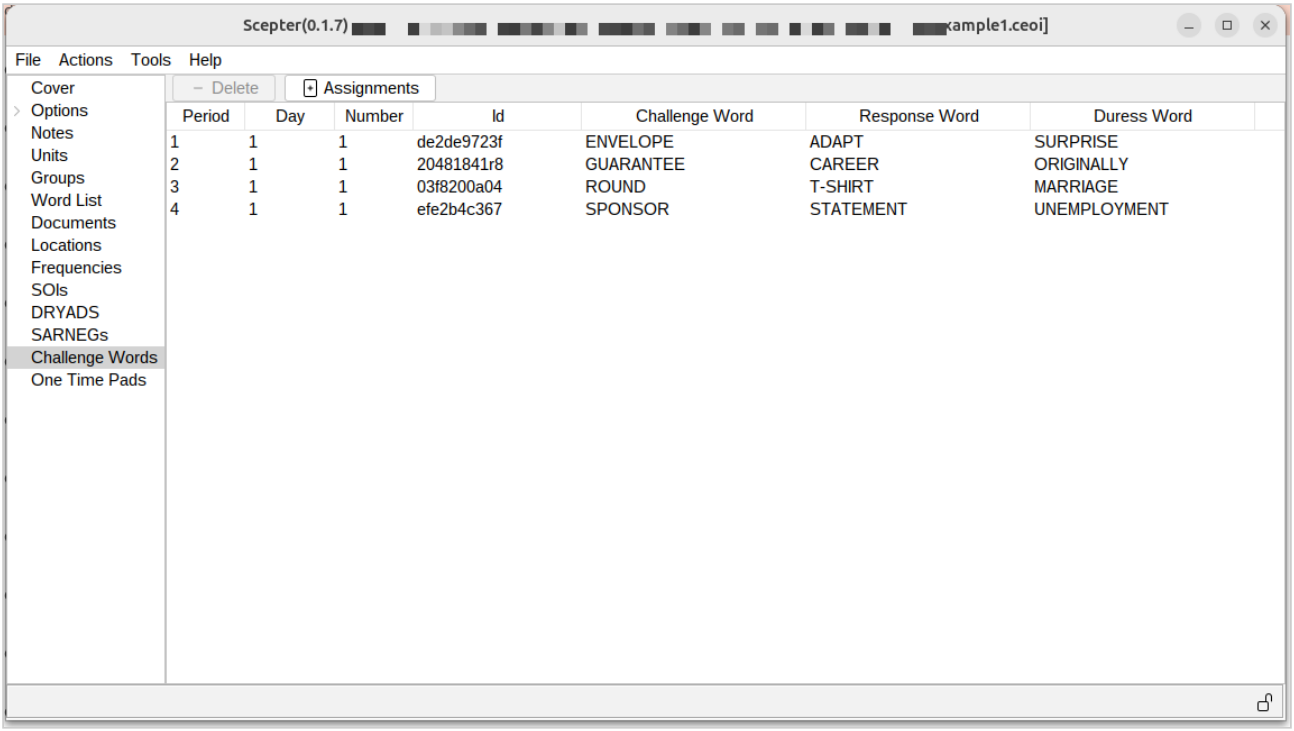
Click **OK** or press **Escape** to close the viewer without making any changes.

> Use **Copy** to quickly paste the SARNEG phonetic word and its mapping into a message or document for distribution.

Related Topics

- SARNEG Panel
- CEOI SARNEGs
- Asset Assignment Dialog

Challenge/Response Words



The Challenge/Response panel displays all challenge and response password pairs in the current CEOI. These are used for identity verification — one party issues a challenge word, and the other must respond with the correct response word to prove their identity.

Challenge/response pairs are generated automatically when you run **Actions > Generate Crypto**. The number and structure depends on the settings in Challenge/Response Settings.

Table Columns

Column	Description
Period	The CEOI period this pair belongs to.
Division	The division (day, week, etc.) within the period. The column header reflects the Division Units setting from Cover Settings.
Number	The sequence number of this pair.
Name	A descriptive name for this pair.
Challenge	The challenge word issued by the verifier.
Response	The correct response word.
Duress	A duress word that signals the responder is under coercion.

Actions

- **Delete** – Select one or more rows and click the **−** button to remove them.

- **Assignments** – Select one or more rows and click the Assignments button to open the Asset Assignment Dialog to assign pairs to units or groups.

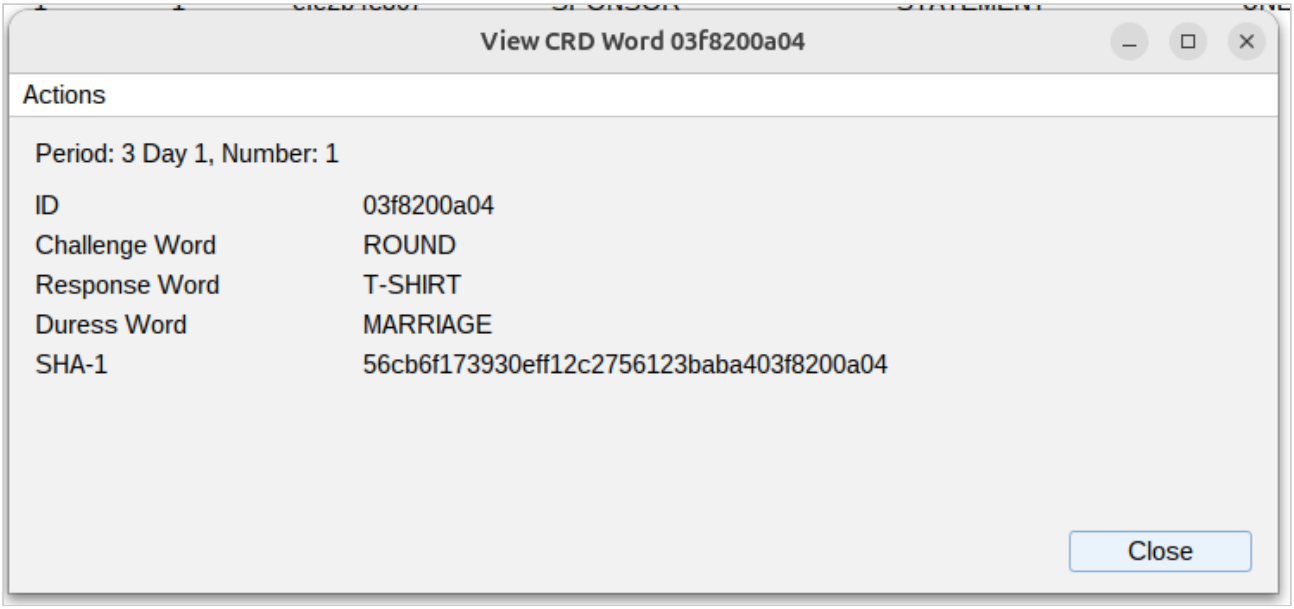
Context Menu

Right-click a row to access:

- **View** – Open the selected pair in the Challenge/Response View Dialog.
- **Print** – Print the selected pairs.
- **Assignments** – Manage unit/group assignments.
- **Change Id...** – Prompts for a new identifier for the selected challenge/response pair and updates it immediately.
- **Delete** – Remove the selected pairs.

> An assignments icon appears in the last column for pairs that have been assigned to specific units or groups.

Challenge/Response/Duress Words Viewer



The CRD Words Viewer displays the details of a single challenge/response/duress word set. It shows the period, division, number, identifier, all three words, and the SHA1 integrity hash. Use this dialog to verify or share the word set, change its identifier, or manage its assignments.

Fields

Field	Description
Period / Division / Number	The CEOI period, division unit, and sequence number this word set belongs to.
ID	The identifier (name) assigned to this challenge/response/duress set.
Challenge	The challenge word — the word that is spoken or sent to request authentication.
Response	The response word — the correct reply to the challenge.
Duress	The duress word — a word that signals the operator is under duress while appearing to authenticate normally.
SHA1	The SHA1 hash of the word set, used to verify data integrity.

Actions Menu

Action	Description
Assignments	Opens the Asset Assignment dialog to manage which units and groups are assigned to this word set.
Copy	

	Copies the challenge, response, and duress words (separated by " / ") to the system clipboard.
Change ID	Prompts for a new identifier and updates it immediately.

Right-Click Menu

The same actions are available by right-clicking anywhere on the dialog.

Closing the Viewer

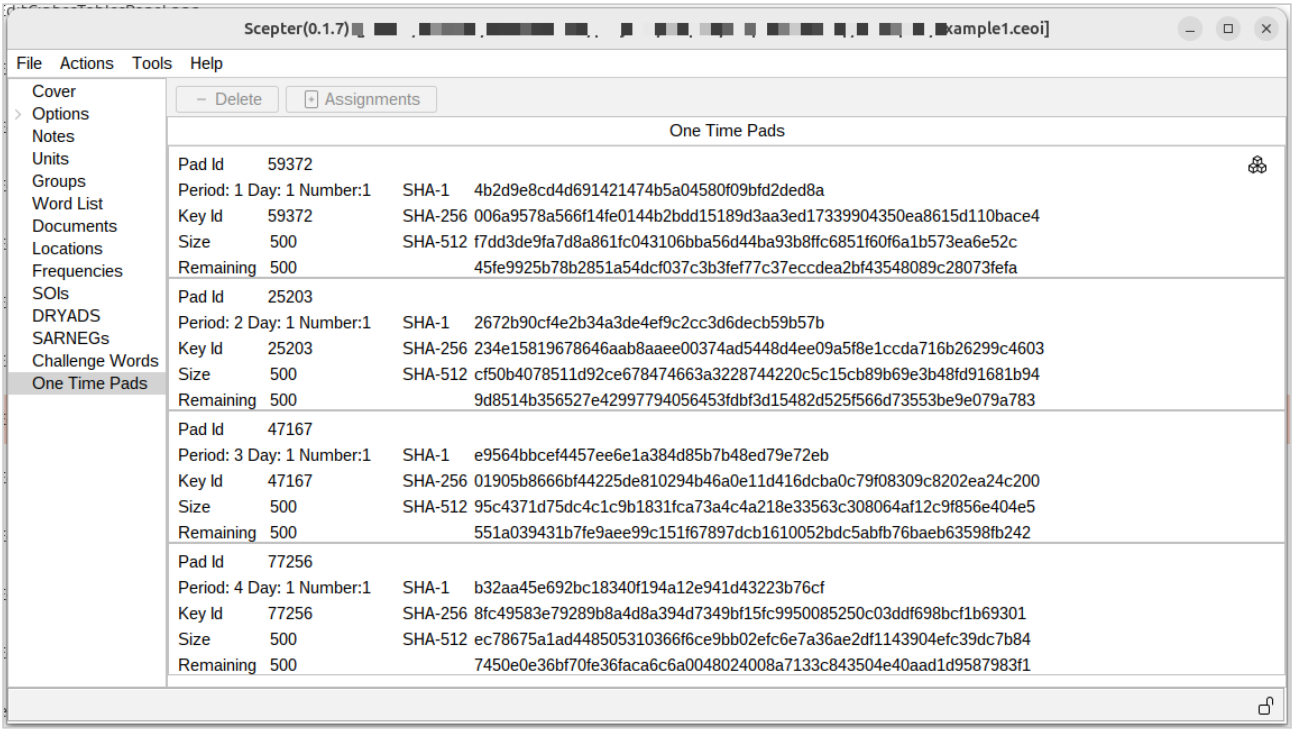
Click **OK** to close the viewer.

> The duress word allows a captured or coerced operator to signal distress while appearing to respond normally. Ensure that duress words are kept confidential and never confused with the response word.

Related Topics

- CEOI Challenge/Response
- Challenge/Response Settings
- Asset Assignment Dialog

One Time Pads



The One Time Pads (OTP) panel displays all OTP sheets in the current CEOI. One-time pads provide theoretically unbreakable encryption when used correctly — each pad is used only once to encrypt a single message.

OTP sheets are generated automatically when you run **Actions > Generate Crypto**. The number and size of generated pads depends on the settings in OTP Settings. You can also import OTPs using **Tools > Import One Time Pad**.

Display

Each OTP is displayed as a compact table cell showing its character grid. Double-click an OTP to open it in the OTP View Dialog for a full-size view.

Actions

- **Delete** – Select one or more OTPs and click the **−** button to remove them.
- **Assignments** – Select one or more OTPs and click the Assignments button to open the Asset Assignment Dialog, where you can assign OTPs to specific units or groups.

Context Menu

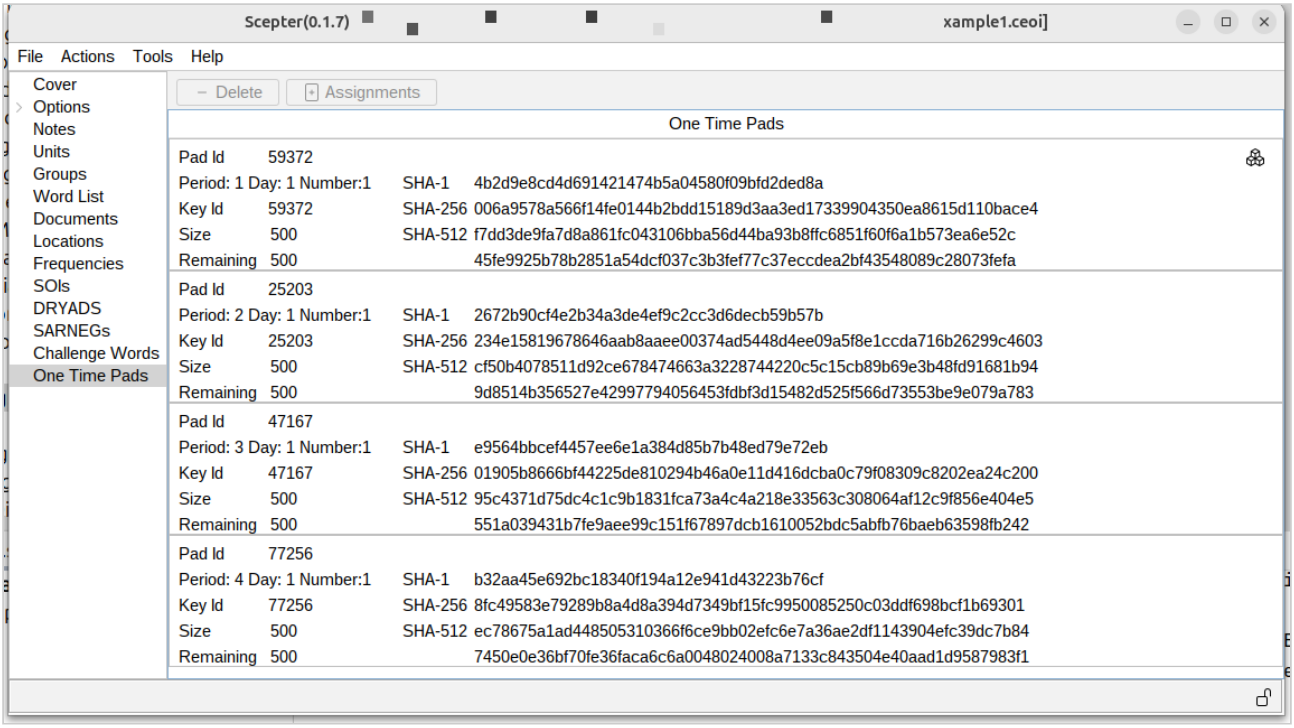
Right-click an OTP to access:

- **View** – Open the OTP in the OTP View Dialog.
- **Assignments** – Manage unit/group assignments.

- **Copy SHA-1 / SHA-256 / SHA-512** – Copy the OTP's hash to the clipboard for verification purposes.
- **Copy Pad** – Copy the OTP character data to the clipboard.
- **Encrypt** – Open the OTP Encrypt dialog to encrypt a message with the selected pad.
- **Decrypt** – Open the OTP Decrypt dialog to decrypt a message with the selected pad.
- **Hash Message** – Open the OTP Hash dialog to create a message hash.
- **Create QR Code** – Generate a QR code containing the OTP data and its SHA-1 hash.
- **Print** – Print the selected OTPs.
- **Delete** – Remove the selected OTPs.

> Each one-time pad must be used only once. Reusing a pad compromises the security of all messages encrypted with it.

OTP Info Panel



The OTP Info Panel is a compact summary view that displays the key properties and integrity hashes for a single one-time pad. It appears as a header in the OTP View Dialog and in the OTP list where each pad is rendered as an info panel cell.

Fields Displayed

Field	Description
Pad Id	The user-assigned identifier for this one-time pad. You can change the Pad Id from the OTP View Dialog using the **Actions > Change Id** menu item.
Period / Division / Number	Shows the pad's position within the CEOI structure -- which period it belongs to, which division (the label reflects the division unit type set on the Cover Settings), and the pad's sequence number within that period and division.
Key Id	The first five characters of the pad's unused portion. The Key Id is used to identify which position in the pad encryption begins from. As characters are consumed, the Key Id advances.
SHA-1	The SHA-1 hash of the pad's character data. Used to verify pad integrity and detect duplicates during import.
SHA-256	The SHA-256 hash of the pad's character data.
SHA-512	The SHA-512 hash of the pad's character data. Displayed across two lines due to its length.
Size	The total number of characters in the pad.

Remaining	The number of unused characters remaining in the pad. As you encrypt messages, consumed characters are replaced with asterisks (`*`) and the remaining count decreases.

Status Indicators

The upper-right area of the panel may display status indicators:

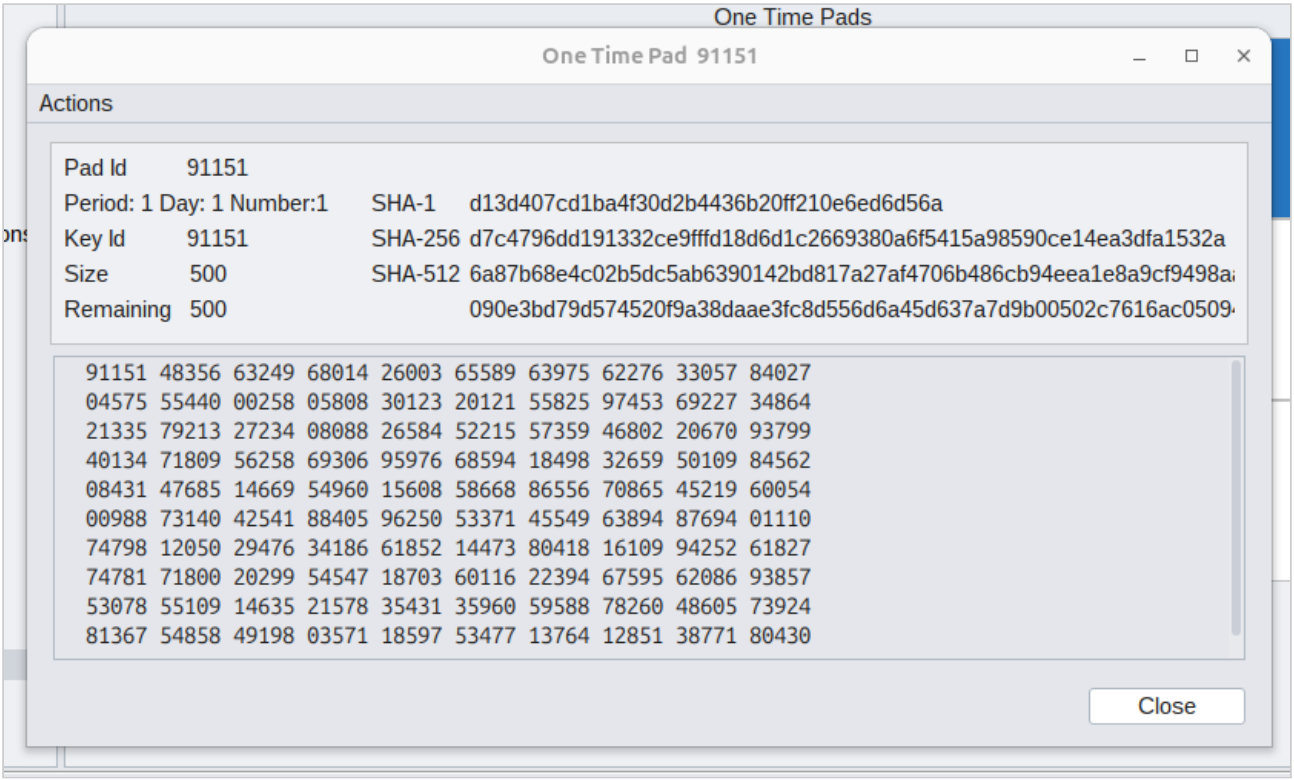
- **Imported** -- Shown when the pad was imported from an external source rather than generated by Scepter.
- **Assignments icon** -- A box icon appears when the pad has been assigned to one or more units or groups via the Asset Assignment Dialog.

> You can use the SHA hashes to verify that a pad received by another party matches the original. Both parties should independently compute the hash and compare results -- if the hashes match, the pad data is identical.

Related Topics

- One Time Pads List
- OTP View Dialog
- OTP Settings

OTP View Dialog



The OTP View Dialog displays the full contents of a one-time pad. It shows the pad's formatted character data in a read-only text area, with an OTP Info Panel header summarizing the pad's metadata and integrity hashes. You can open this dialog by double-clicking an OTP in the OTP list, or by right-clicking an OTP and selecting **View**.

Actions Menu

The **Actions** menu at the top of the dialog provides the following operations:

Menu Item	Description
Copy SHA-1 to Clipboard	Copies the pad's SHA-1 hash string to the system clipboard.
Copy SHA-256 to Clipboard	Copies the pad's SHA-256 hash string to the system clipboard.
Copy SHA-512 to Clipboard	Copies the pad's SHA-512 hash string to the system clipboard.
Copy Pad to Clipboard	Copies the entire formatted pad data to the system clipboard.
QR Code	Generates a QR code containing the pad data followed by its SHA-1 hash. This allows the pad to be transferred optically to another device or printed for scanning.

Change Id	Opens a dialog to change the pad's identifier. Enter the new Pad Id and click OK to apply.
Encrypt Message	Opens the OTP Encrypt Dialog with this pad pre-selected. This option is hidden in view-only mode.
Decrypt Message	Opens the OTP Decrypt Dialog with this pad pre-selected. This option is hidden in view-only mode.
Message Hash	Opens the Message Hash Tool for generating salted hashes.

Context Menu

Right-click anywhere in the pad text area to access:

- **Copy to Clipboard** -- Copies the visible pad text to the clipboard.

View-Only Mode

When opened from certain contexts (such as the Message Encrypt/Decrypt Dialog via the view pad button), the OTP View Dialog opens in view-only mode. In this mode, the **Encrypt Message** and **Decrypt Message** menu items are hidden because the pad is being viewed for reference only during an active encryption or decryption operation.

> Use the QR Code feature to transfer a pad to a mobile device or another workstation. The generated QR code includes the SHA-1 hash so the receiving party can verify the pad was received correctly.

> The pad data is displayed in a monospaced font with formatted spacing for readability. Characters that have been consumed by previous encryption operations appear as asterisks (`*`).

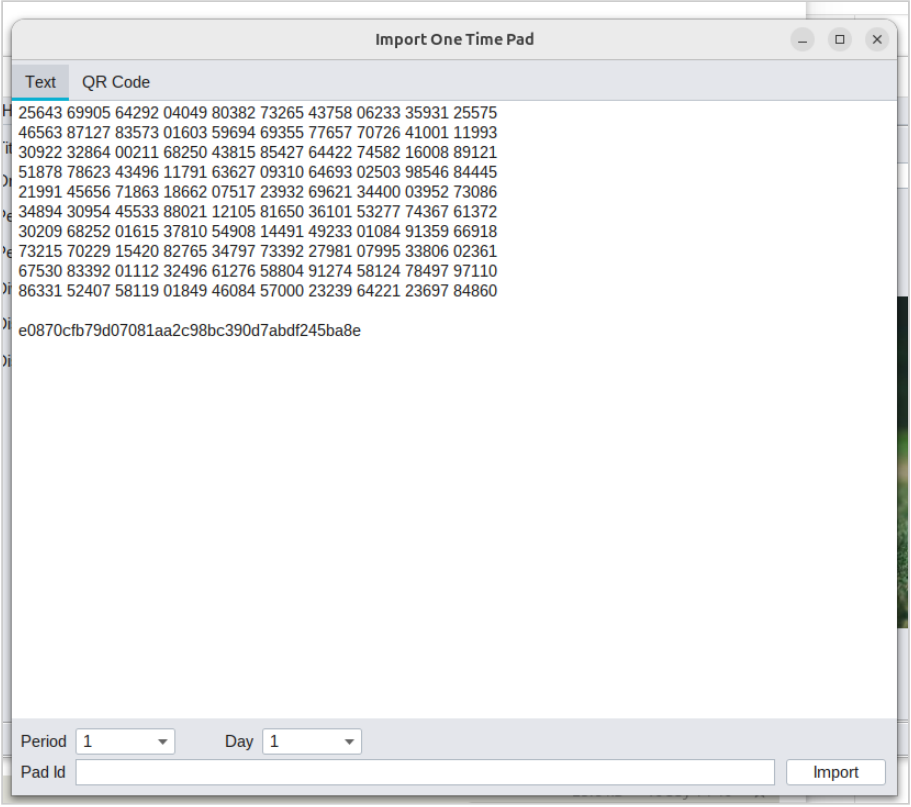
Keyboard Shortcuts

Key	Action
`Escape`	Close the dialog

Related Topics

- One Time Pads List
- OTP Info Panel
- OTP Encrypt Dialog
- OTP Decrypt Dialog
- Message Hash Tool
- OTP Settings

Import One Time Pad



The Import One Time Pad dialog allows you to bring externally generated one-time pads into the current CEOI. This is useful when pads are created by another system or shared between parties. You can access this dialog from **Tools > Import One Time Pad** in the main menu.

Input Tabs

The dialog provides two tabs for entering pad data:

Tab	Description
Text	A text area where you can type or paste the raw pad characters. Spaces and newlines are automatically stripped during import. All characters are converted to uppercase.
QR Code	A QR code scanner panel that decodes a QR code image containing pad data. When a QR code is successfully decoded, the data is transferred to the Text tab automatically.

Import Settings

Below the text input area, you must specify the following fields before importing:

Field	Description
Period	The CEOI period number to assign the imported pad to. The dropdown shows all available periods defined in the current CEOI.

Division	The division number to assign the imported pad to. The label for this field reflects the division unit type set on the Cover Settings (for example, "Team" or "Squad"). The dropdown shows all available divisions defined in the current CEOI.
Pad Id	An optional identifier for the imported pad. If left blank, the pad's Key Id (the first five unused characters) is used as the Pad Id.

How to Import

1. Enter or paste the pad characters into the **Text** tab, or scan a QR code on the **QR Code** tab.
2. Select the target **Period** and **Division** for the pad.
3. Optionally enter a **Pad Id** to identify the pad.
4. Click **Import**.

If the import succeeds, a result dialog displays the imported pad's details including its computed hashes. The text area is cleared so you can import additional pads.

Pad Type Detection

The pad type is detected automatically from the characters in the data:

- If the data contains only digits (0-9), the pad type is set to **Numbers**.
- If the data contains only uppercase letters (A-Z), the pad type is set to **DIANA** (Letters).
- If the data contains both letters and digits, the pad type is set to **DIANA+** (Numbers and Letters).

Validation

The importer checks the following conditions and displays an error if any fail:

- The pad must contain at least 10 characters.
- Only uppercase letters (A-Z) and digits (0-9) are valid. Any other character causes an error.
- The pad must not duplicate an existing pad in the CEOI. Duplicates are detected by comparing SHA-1 hashes.

> Imported pads are marked with an "Imported" status indicator in the OTP Info Panel and the OTP list. This distinguishes them from pads generated by Scepter.

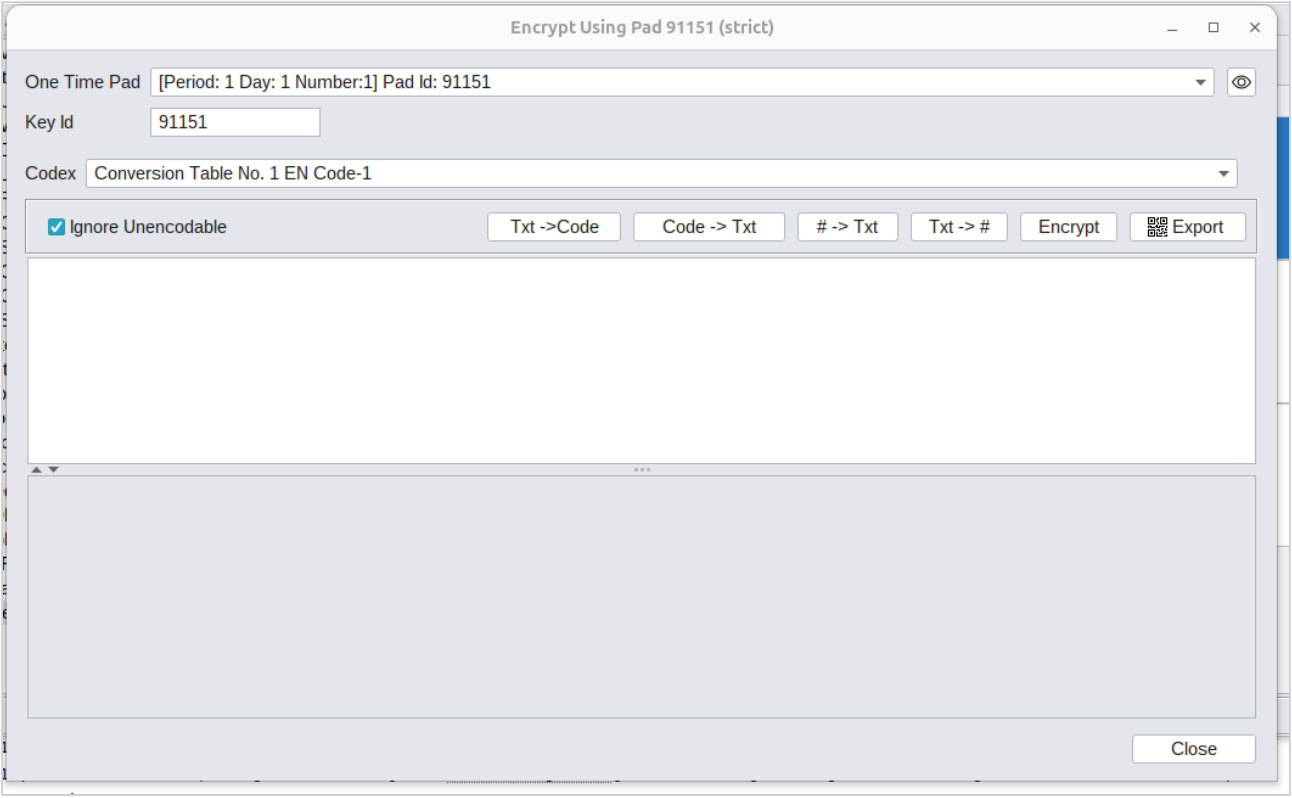
> You can use the QR Code tab to scan a QR code exported from another Scepter installation or from the OTP View Dialog's QR Code feature. The QR code format includes the pad data followed by its SHA-1 hash for verification.

Related Topics

- One Time Pads List
- OTP View Dialog
- OTP Info Panel

- OTP Settings

OTP Encrypt Dialog



The OTP Encrypt Dialog encrypts a plaintext message using a one-time pad. It is a mode of the OTP Message Dialog that opens when you select **Encrypt** from the OTP list context menu or from the OTP View Dialog's Actions menu.

How to Encrypt a Message

1. Select the one-time pad to use from the **One Time Pad** dropdown at the top. The pad's Key Id is automatically populated in the **Key Id** field.
2. If using a numeric pad, select the appropriate **Codex** from the options panel. If using a DIANA or DIANA+ pad, configure the space character option if needed.
3. Type or paste your plaintext message into the upper text area.
4. Optionally use the toolbar buttons to pre-process the message:
    - **Txt -> Code** -- Encodes the message using the CEOI word list, replacing words with their assigned codes.
    - **Txt -> #** -- Converts text characters to their numeric equivalents.
5. Click **Encrypt**.
6. The encrypted cipher text appears in the lower (output) text area.

Toolbar Reference

Button	Description

Ignore Unencodable	When checked (the default), characters that cannot be represented in the pad's character set are silently skipped during encryption. When unchecked, encountering an unencodable character produces an error.
Txt -> Code	Encodes the input text using the CEOI word list. Each recognized word or phrase is replaced with its assigned code. The word list period and division are determined by the selected pad's period and division, or you are prompted to select them if the word list generation mode differs from the OTP generation mode.
Code -> Txt	Decodes word list codes in the input text back into their plaintext equivalents.
# -> Txt	Converts numbers in the input text to their text representation.
Txt -> #	Converts text in the input to its numeric representation.
Encrypt	Performs the OTP encryption using the selected pad and writes the result to the output area.
Export (QR Code)	Exports the cipher text from the output area as a QR code for optical transfer.

Pad Type Behavior

Numeric Pads

When encrypting with a numeric pad, you must select a **Codex** conversion table. The codex maps each letter to a two-digit number. The plaintext is first converted to numbers using the codex, and then each digit is added to the corresponding pad digit modulo 10 to produce the cipher text.

DIANA Pads (Letters)

DIANA encryption uses a trigraph table to combine each plaintext letter with the corresponding pad letter. A **Space character** option appears -- you can specify a letter to substitute for spaces in the plaintext. The **Automatic space selection** checkbox finds an unused letter in the message to use as the space character automatically.

DIANA+ Pads (Numbers and Letters)

DIANA+ works like DIANA but supports both letters and digits in the pad and message. The same space character options are available.

Context Menus

Right-click in the input (top) text area to access:

- **Copy** -- Copies the input text to the clipboard.
- **Paste** -- Pastes text from the clipboard into the input area.
- **Import from QR Code** -- Scans and decodes a QR code image to import text into the input area.

Right-click in the output (bottom) text area to access:

- **Copy** -- Copies the cipher text to the clipboard.
- **Export to QR Code** -- Generates a QR code from the cipher text.

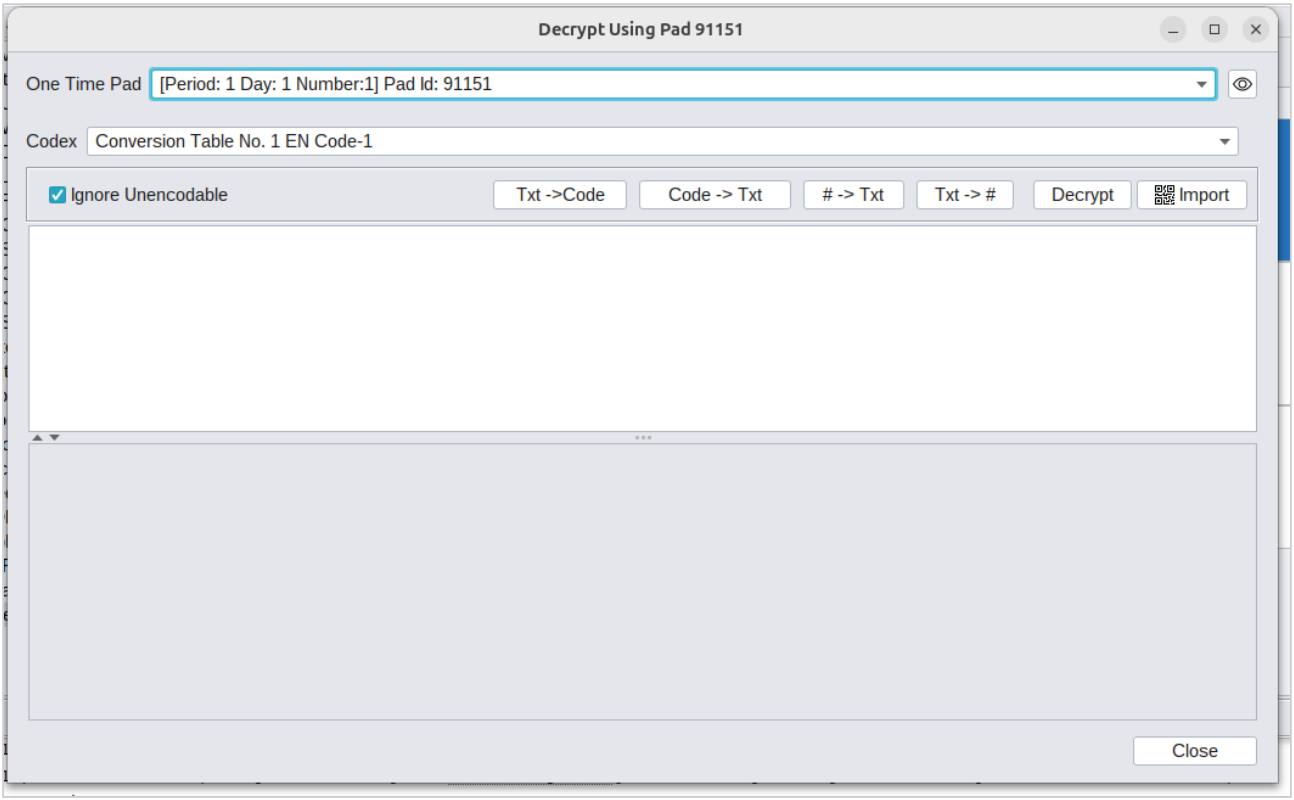
> Use the word list encoding step (**Txt -> Code**) before encrypting to reduce the message length. Word list codes are typically shorter than the words they represent, which uses less of the pad.

> Each section of a one-time pad must be used only once. In strict mode, the pad characters used for encryption are consumed and replaced with asterisks (`*`). Never reuse pad material -- doing so compromises the security of both the original and new messages.

Related Topics

- OTP Decrypt Dialog
- OTP Message Dialog Overview
- One Time Pads List
- OTP View Dialog
- OTP Settings
- Message Hash Tool

OTP Decrypt Dialog



The OTP Decrypt Dialog decrypts a cipher text message using a one-time pad. It is a mode of the OTP Message Dialog that opens when you select **Decrypt** from the OTP list context menu or from the OTP View Dialog's Actions menu.

How to Decrypt a Message

1. Select the one-time pad that was used to encrypt the message from the **One Time Pad** dropdown.
2. If using a numeric pad, select the same **Codex** that was used during encryption. If using a DIANA or DIANA+ pad, configure the space character option to match the encryption settings.
3. Enter the cipher text into the upper text area. You can type it, paste it, or import it from a QR code.
4. Click **Decrypt**.
5. The decrypted plaintext appears in the lower (output) text area.

Toolbar Reference

Button	Description
Ignore Unencodable	When checked, characters that cannot be processed are silently skipped during decryption.
Txt -> Code	Encodes text in the input area using the CEOI word list. Less commonly used during decryption, but available if you need to look up codes.
Code -> Txt	

	Decodes word list codes in the input area back into their plaintext equivalents. Use this after decryption if the original message was word-list-encoded before encryption.
# -> Txt	Converts numbers in the input text to their text representation.
Txt -> #	Converts text in the input to its numeric representation.
Decrypt	Performs the OTP decryption using the selected pad and writes the plaintext result to the output area.
Import (QR Code)	Scans and decodes a QR code image to import cipher text into the input area.

Differences from Encrypt Mode

- The **Key Id** field is hidden in decrypt mode. The decryption process locates the correct position in the pad automatically or based on the cipher text content.
- The QR Code button is labeled **Import** instead of **Export**, and it scans a QR code to populate the input area with cipher text rather than exporting output.
- The **Encrypt** button is replaced by a **Decrypt** button.

Post-Decryption Steps

After decrypting a message, you may need to reverse any pre-processing that was applied before encryption:

- If the sender used word list encoding, copy the decrypted output to the input area and click **Code -> Txt** to decode the word list codes back to plaintext.
- If the sender used numeric text conversion, use **# -> Txt** to convert numbers back to text.

DIANA / DIANA+ Automatic Space Recovery

When decrypting a DIANA or DIANA+ message, if the **Automatic space selection** checkbox is enabled, the decryptor examines the first character of the decrypted output and uses it as the space substitution character. All occurrences of that character in the decrypted text are replaced with actual spaces, and leading/trailing spaces are trimmed. This reverses the automatic space selection that was applied during encryption.

Context Menus

Right-click in the input (top) text area to access:

- **Copy** -- Copies the input text to the clipboard.
- **Paste** -- Pastes text from the clipboard into the input area.
- **Import from QR Code** -- Scans and decodes a QR code image to import cipher text.

Right-click in the output (bottom) text area to access:

- **Copy** -- Copies the decrypted text to the clipboard.

- **Export to QR Code** -- Generates a QR code from the output text.

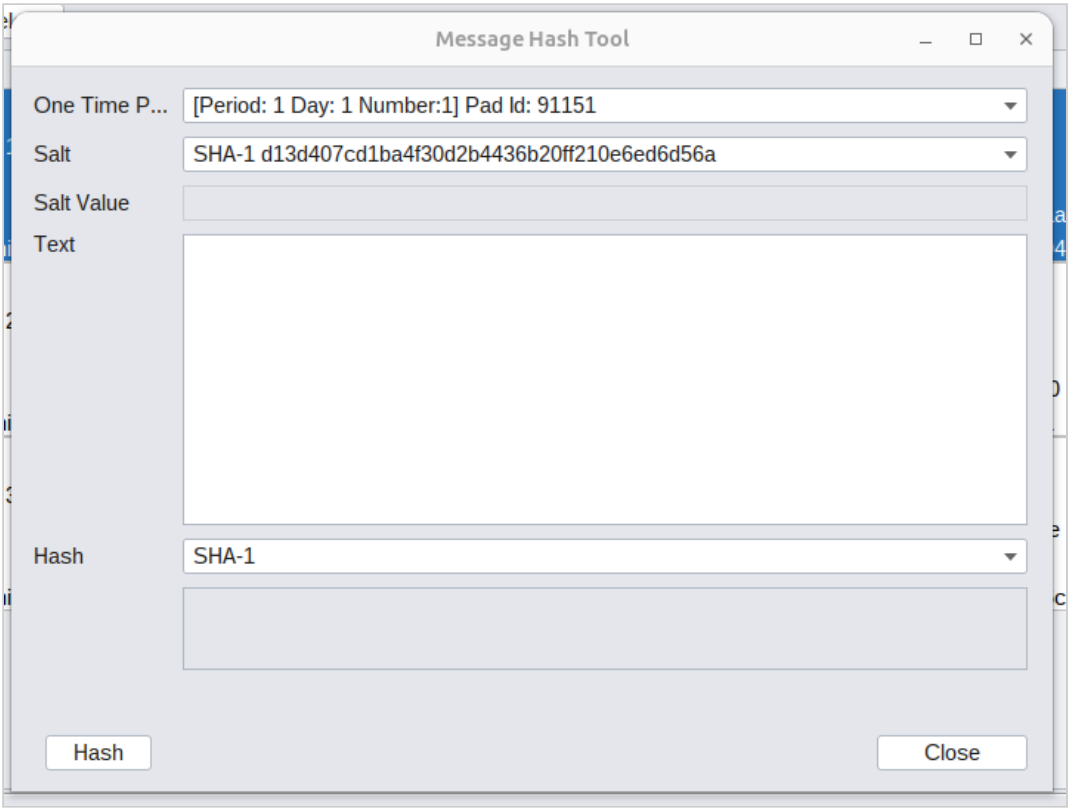
> When receiving an encrypted message, make sure you use the same pad and the same codex (for numeric pads) or the same space character setting (for DIANA pads) that the sender used during encryption. If these do not match, decryption will produce garbled output.

> In strict mode, pad characters are consumed during decryption and replaced with asterisks (`*`). The dialog title shows "(strict)" when strict mode is active.

Related Topics

- OTP Encrypt Dialog
- OTP Message Dialog Overview
- One Time Pads List
- OTP View Dialog
- OTP Settings
- Message Hash Tool

Message Hash Tool



The Message Hash Tool generates cryptographic hashes of text, optionally salted with a one-time pad, a DRYAD, or a custom value. Message hashes can be used to verify message integrity -- the sender and receiver independently hash the message with the same salt, and if the hashes match, the message was not altered in transit. You can open this dialog from the OTP list context menu (**Hash Message**), from the OTP View Dialog's Actions menu, from the DRYADs context menu, or from the DRYAD Viewer's Actions menu.

Fields Reference

Field	Description
Salt Source	Selects whether the salt is derived from a **One Time Pad**, a **DRYAD**, or a custom (**Other**) value. The available options depend on which items exist in the current CEOI. If neither OTPs nor DRYADs are present, this selector is hidden and a custom salt value must be entered directly.
One Time Pad / DRYAD	When **One Time Pad** is selected as the source, this dropdown lists the available OTPs. When **DRYAD** is selected, it lists the available DRYADs. The chosen item is used to derive the salt value.
Salt	Selects what value to use as the salt. The salt is prepended to the message text before hashing. Options include: • **SHA-1** -- Uses the selected pad's SHA-1 hash as the salt. • **SHA-256** -- Uses the selected pad's SHA-256 hash as the salt.

	• **SHA-512** -- Uses the selected pad's SHA-512 hash as the salt. • **Entire Pad** -- Uses the full formatted pad data as the salt. • **Other** -- Enables the Salt Value text field so you can enter a custom salt.
Salt Value	A text area for entering a custom salt value. This field is only enabled when the Salt dropdown is set to **Other**, or when no OTPs are available.
Text	The message text to hash. Enter or paste the message content here. The salt value is prepended to this text before the hash is computed.
Hash	Selects the hash algorithm to use. Options are: • **SHA-1** -- Produces a 160-bit hash. • **SHA-256** -- Produces a 256-bit hash. • **SHA-512** -- Produces a 512-bit hash. • **MD5 Sum** -- Produces a 128-bit hash. • **APRS Call Sign Hash** -- Produces a numeric APRS passcode from a callsign string. The salt value is not used for this hash type.

How to Generate a Hash

1. Select a **Salt Source**: **One Time Pad**, **DRYAD**, or **Other** (custom).
2. Choose the specific OTP or DRYAD from the second dropdown, or type your salt value in the **Salt Value** field if **Other** was selected.
3. Choose a **Salt** hash variant (SHA-1, SHA-256, SHA-512, or Entire Pad/Table) from the Salt dropdown.
4. Enter the message text in the **Text** field.
5. Select the desired **Hash** algorithm.
6. Click **Hash**.
7. The resulting hash appears in the output area below the algorithm selector.

Copying the Hash

Right-click the hash output area and select **Copy to Clipboard** to copy the hash string.

APRS Call Sign Hash

The **APRS Call Sign Hash** option generates a numeric APRS passcode from a callsign entered in the Text field. This is a specialized hash used in the Automatic Packet Reporting System. Unlike the other hash algorithms, the APRS hash does not use a salt value -- only the callsign text in the Text field is used as input.

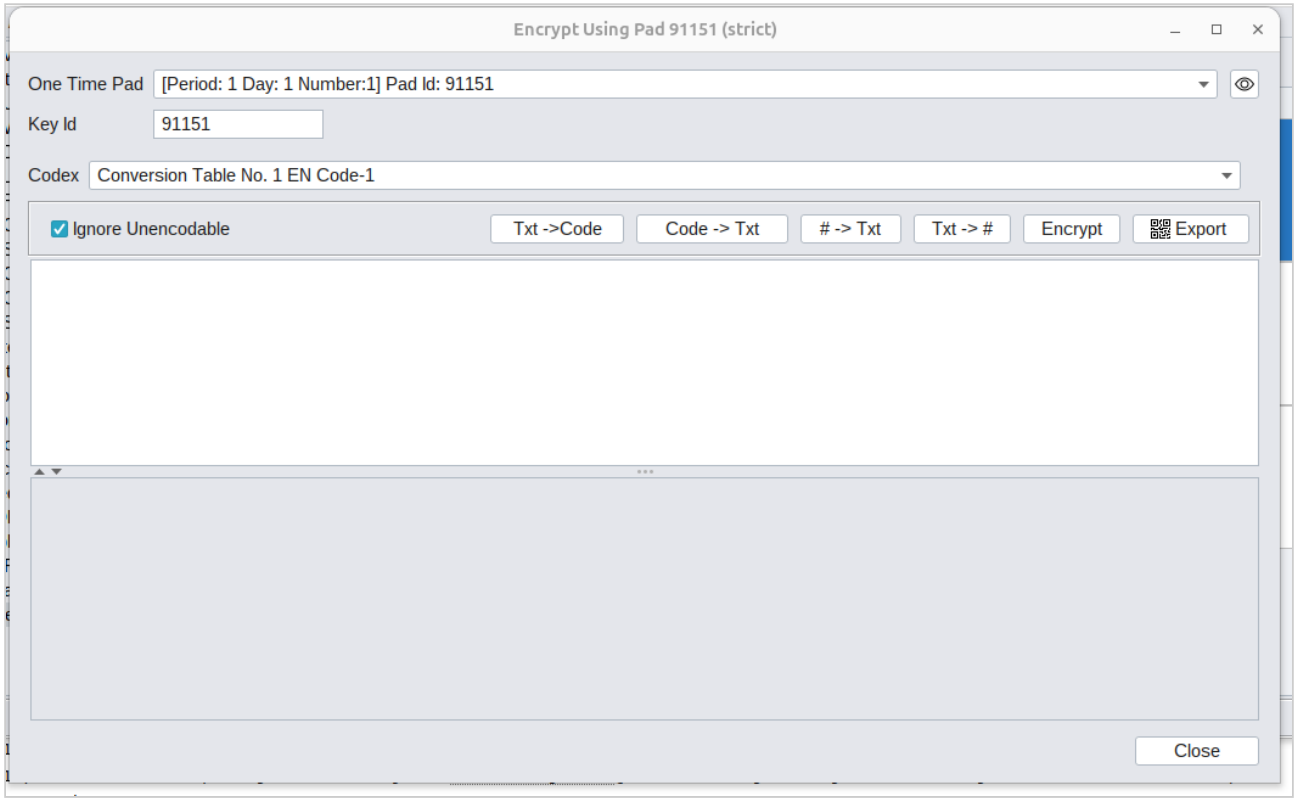
To verify message integrity between two parties: both parties must agree on the same OTP (or custom salt) and hash algorithm, then independently hash the message. If both hashes match, the message content is confirmed identical.

The hash computation concatenates the salt value and the message text (salt + text), then applies the selected hash algorithm to the combined string. Both parties must use the same salt and the same algorithm to produce matching hashes.

Related Topics

- One Time Pads List
- DRYADs
- DRYAD Viewer
- OTP View Dialog
- OTP Encrypt Dialog
- OTP Decrypt Dialog
- OTP Settings

OTP Message Encrypt / Decrypt



The OTP Message Dialog is a dual-purpose dialog used for encrypting and decrypting messages with a one-time pad. It opens in either Encrypt or Decrypt mode depending on how you access it. Both modes share the same layout: a split-pane interface with an input text area on top and an output text area on the bottom, separated by a toolbar of actions.

Accessing the Dialog

You can open this dialog from several locations:

- Right-click an OTP in the OTP list and select **Encrypt** or **Decrypt**.
- From the OTP View Dialog, use **Actions > Encrypt Message** or **Actions > Decrypt Message**.

Dialog Layout

Area	Description
One Time Pad selector	A dropdown at the top of the dialog listing all OTPs in the current CEOI. Changing the selection switches the active pad. A view button (eye icon) opens the selected pad in the OTP View Dialog in view-only mode.
Key Id	Shows (and allows editing of) the key identifier used for encryption. The Key Id is the first five unused characters of the pad and tells the recipient where in the pad to start decryption. This field is visible only in encrypt mode.
Options panel	

	Additional options that appear based on the pad type. For numeric pads, a Codex selector appears. For DIANA and DIANA+ pads, space character options appear. See the pad-type-specific help pages for details.
Input area (top pane)	Where you enter the message to encrypt, or the cipher text to decrypt.
Toolbar	Contains the action buttons: word list encoding/decoding, number/text conversion, encrypt/decrypt, and QR code import/export.
Output area (bottom pane)	Displays the result -- cipher text after encryption, or plaintext after decryption. This area is read-only.

Pad Type Options

The dialog dynamically shows additional options depending on the type of the selected one-time pad:

- **Numbers** -- A **Codex** dropdown appears, allowing you to select which codex conversion table to use for translating between letters and numbers. See OTP Settings for information on codex tables.
- **DIANA / DIANA+** -- A **Space character** field and **Automatic space selection** checkbox appear. The space character is a letter substituted for spaces in the plaintext before encryption. When automatic selection is enabled, the system chooses an unused character from the cipher alphabet.

Strict Mode

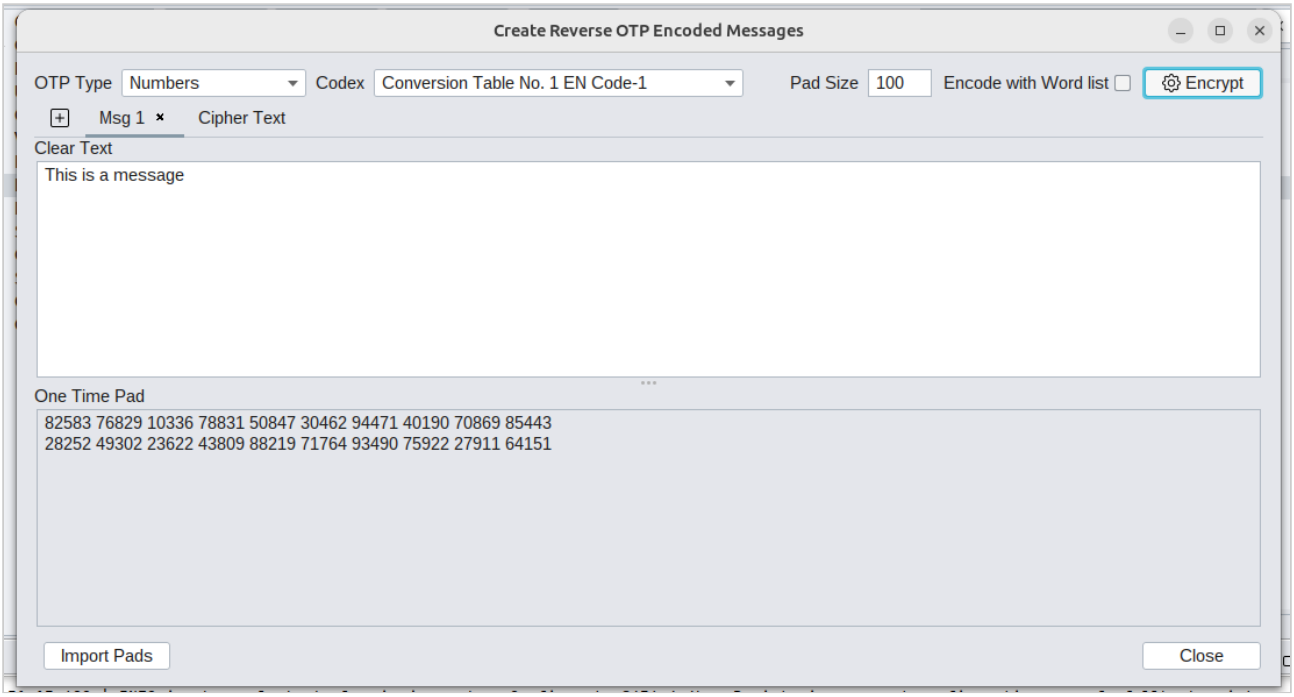
The dialog title indicates whether strict mode is active, shown as "(strict)" after the pad name. In strict mode, pad characters are consumed after use and cannot be reused. This enforces proper one-time pad security practice.

> For detailed instructions on encryption, see OTP Encrypt Dialog. For decryption, see OTP Decrypt Dialog.

Related Topics

- OTP Encrypt Dialog
- OTP Decrypt Dialog
- One Time Pads List
- OTP View Dialog
- Message Hash Tool
- OTP Settings

Reverse OTP Message



The Reverse OTP Message dialog lets you encrypt one or more plain-text messages simultaneously into a single shared cipher text, and generates a unique one-time pad (OTP) for each message. The "reverse" technique produces a shared ciphertext that decrypts to different plaintexts depending on which OTP is applied — allowing multiple recipients to each receive a different plaintext from the same ciphertext.

OTP Type and Codex

Select the OTP type to use for encryption:

- **Letters** — the OTP and ciphertext use alphabetic characters.
- **Numbers** — the OTP and ciphertext use numeric characters. When Numbers is selected you can also choose a **Codex** to define the letter-to-digit mapping.

Minimum OTP Size

Enter the minimum number of characters for the generated OTP. Leave blank to use the minimum required by the message content.

Use Word List

Check **Use Word List** to pre-encode each message using the CEOI word list before encryption. When selected, you will be prompted to choose the period and division. This option is only available when the CEOI contains word list entries.

Messages

Each message is entered on its own tab. Use the **+** tab icon to add a new message tab. Each tab has a label (e.g., "Msg 1", "Msg 2") and contains:

- A text area for the plain-text message.
- An OTP area that is populated with the generated one-time pad after encryption.

Click the **×** button on a tab to remove it. You need at least one message tab to encrypt.

Cipher Text Tab

The **Cipher Text** tab shows the shared ciphertext generated after encryption. Right-click the cipher text to copy it to the clipboard.

Encrypting

1. Add message tabs and enter the plain text for each recipient.
2. Select OTP type, codex (if numeric), and optional minimum OTP size.
3. Click **Encrypt**. The cipher text is generated and individual OTPs appear in each message tab.

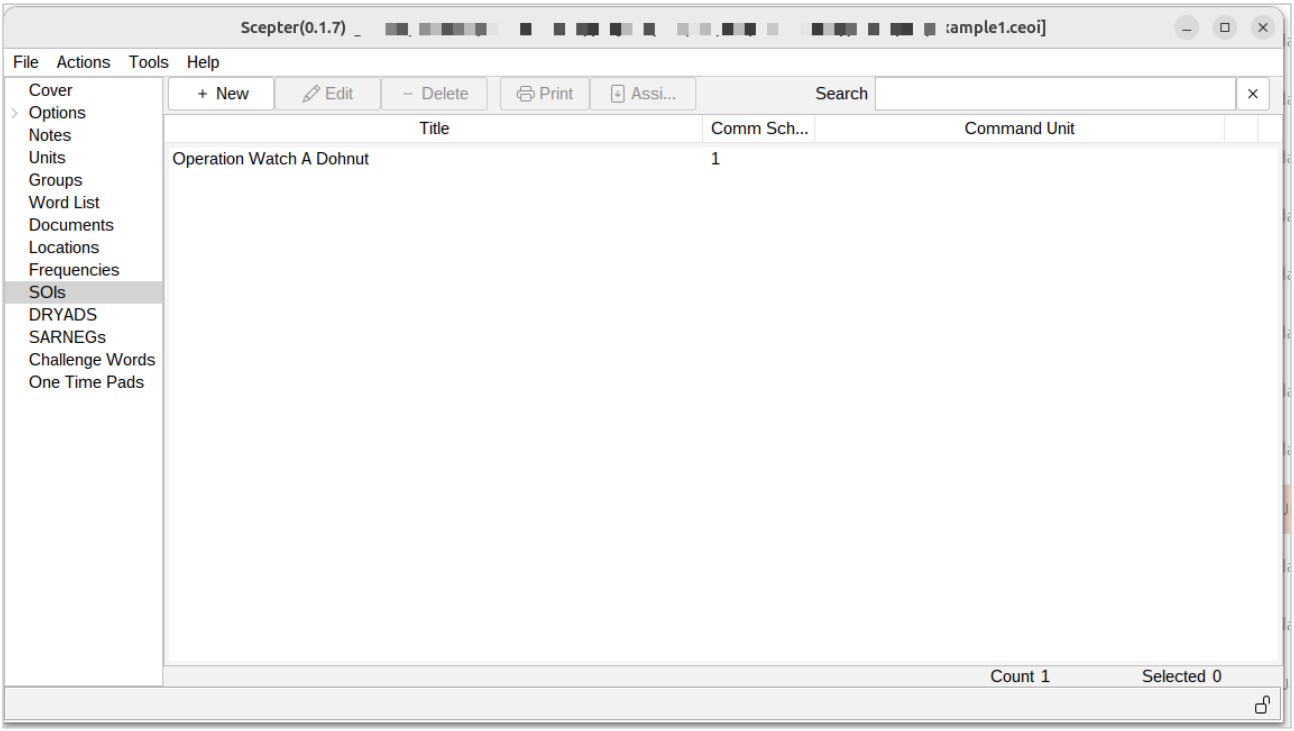
Importing OTPs

After encryption, click **Import Pads** to import the generated OTPs into the CEOI one-time pad store. You will be prompted to select the period and division. Each pad is named using the message label and a hash suffix.

Related Topics

- OTP Message Encrypt
- OTP Message Decrypt
- CEOI One Time Pads
- Word List Encoder/Decoder

Signal Operating Instructions (SOIs)



The SOI panel displays all Signal Operating Instructions defined in the current CEOI. An SOI is a structured communications plan that defines frequency assignments, schedules, control units, and PACE (Primary, Alternate, Contingency, Emergency) signal plans for a specific operational time window. SOIs are used to coordinate radio communications across participating units during a defined period.

Table Columns

Column	Description
Title	The descriptive title of the SOI.
Duration	The duration of the SOI in days.
Command Unit	The name of the control unit responsible for this SOI. Displays the unit name looked up from the CEOI's unit list.
Assets	The number of assets issued to this SOI. See the Assets tab of the SOI Edit Dialog.
(Icon)	A small icon appears in this column if this SOI has been assigned to any units or groups.

Actions

- **Add** – Click the **+** button or press `Insert` to create a new SOI. The SOI Edit Dialog opens for you to define the SOI's details, including title, frequencies, PACE plan, unit assignments, communication schedule, and SARNEG.

- **Edit** – Select an SOI and click the pencil button, press `Enter`, or double-click the row to open the SOI Edit Dialog for the selected SOI.
- **Delete** – Select one or more SOIs and click the **−** button or press `Delete` to remove them from the CEOI. Deleting an SOI also withdraws it from every unit and group it was issued to, and releases the assets it held.
- **Print** – Select one or more SOIs and click the printer button to print the selected SOIs using the configured print template.
- **Assignments** – Select one or more SOIs and click the Assignments button to open the Asset Assignment Dialog, where you can assign the selected SOIs to specific units or groups. This button is only enabled when at least one SOI is selected.
- **Print Assets** – Select one or more SOIs and click the Print Assets button to open the Asset Print Dialog for each, which prints the assets issued to that SOI using the template assigned to each one. This button is only enabled when at least one SOI is selected.

Right-Click Context Menu

Right-clicking on the table displays a context menu with the same actions available from the toolbar: New, Edit, Delete, Assignments, Print Assets, Print, Import List, and Export List.

Search

Type in the search field above the table to filter the list. Clear the search field to show all SOIs again.

Import and Export

You can import SOIs from a file or export the current SOI list using the **Tools > Import List** and **Tools > Export List** menu items when this panel is active.

SOI Details

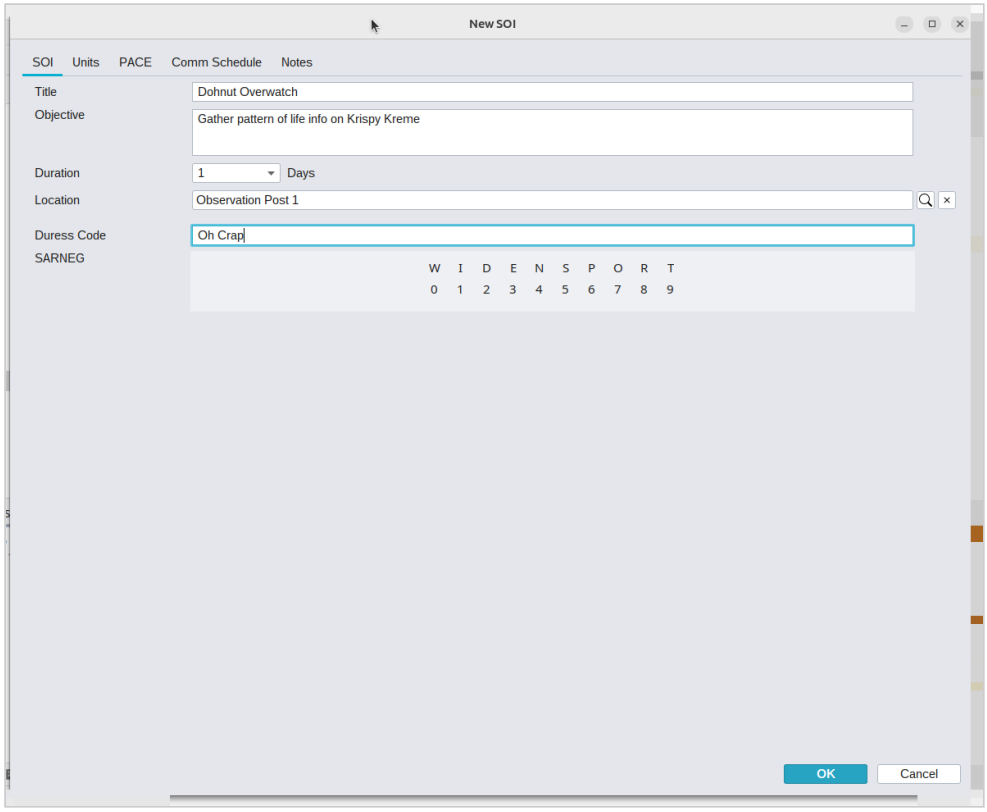
Each SOI contains the following information, which you configure in the SOI Edit Dialog:

- **Title and Objective** – A descriptive title and mission objective for the SOI.
- **Duration** – The number of days the SOI is in effect.
- **PACE Frequencies** – Primary, Alternate, Contingency, and Emergency transmit/receive frequency assignments with associated prowords and signal methods.
- **Control and Recovery Units** – The designated control unit and recovery unit, with their callsigns.
- **Location** – An optional location reference for the SOI.
- **Unit Assignments** – A list of units participating in the SOI, with assigned roles.
- **Communication Schedule** – A schedule of communication windows defining when contacts are expected.
- **SARNEG** – An associated SARNEG table for authentication.
- **Duress Code** – A code used to indicate communication under duress.
- **Asset Assignments** – Assignments of this SOI to units or groups for report generation.

The status bar at the bottom of the panel shows the total number of SOIs and how many are currently selected.

SOI generation settings such as default duration, per-day SARNEG and password options, and the default print template can be configured on the SOI Settings sub-panel under Options.

SOI Edit Dialog



The SOI (Signal Operating Instructions) Edit Dialog is used to create or edit an SOI entry. An SOI defines the communications plan for an operation, including the PACE (Primary, Alternate, Contingency, Emergency) plan, unit assignments, communication schedule, and associated assets.

Tabs

Tab	Description
SOI	Core SOI details: title, objective, duration, location, duress code, and SARNEG. See SOI Details.
Units	Assign control and recovery units, plus additional participating units. See SOI Units.
PACE	Configure the Primary, Alternate, Contingency, and Emergency communications plan. See SOI PACE Plan.
Comm Schedule	Define communication windows for each day of the SOI duration. See SOI Comm Schedule.
Notes	Free-text notes about this SOI. See SOI Notes.
Assignments	The units and groups this SOI is issued to. See SOI Assignments.
Assets	The assets issued to this SOI, and the template each prints with. See SOI Assets.

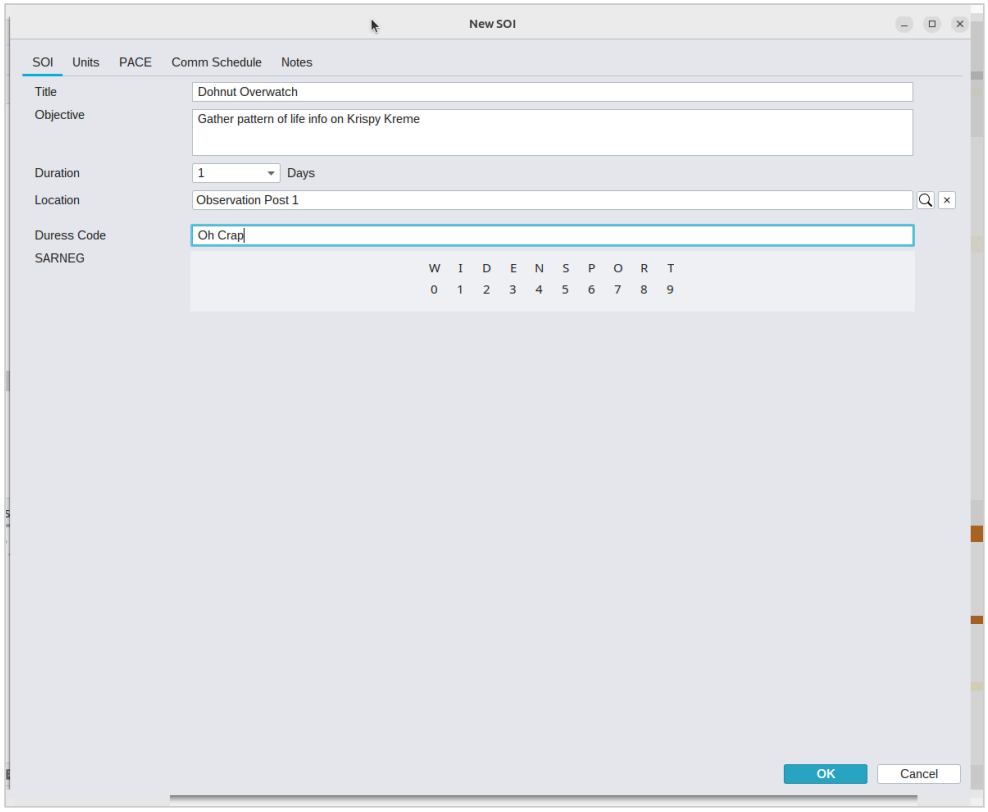
Actions

Button	Description
OK	Validates the form, saves changes, and closes the dialog. The Title field is required.
Cancel	Discards changes and closes the dialog. Press `Escape` to cancel.

Related Topics

- SOI Panel
- SOI Settings

SOI Details



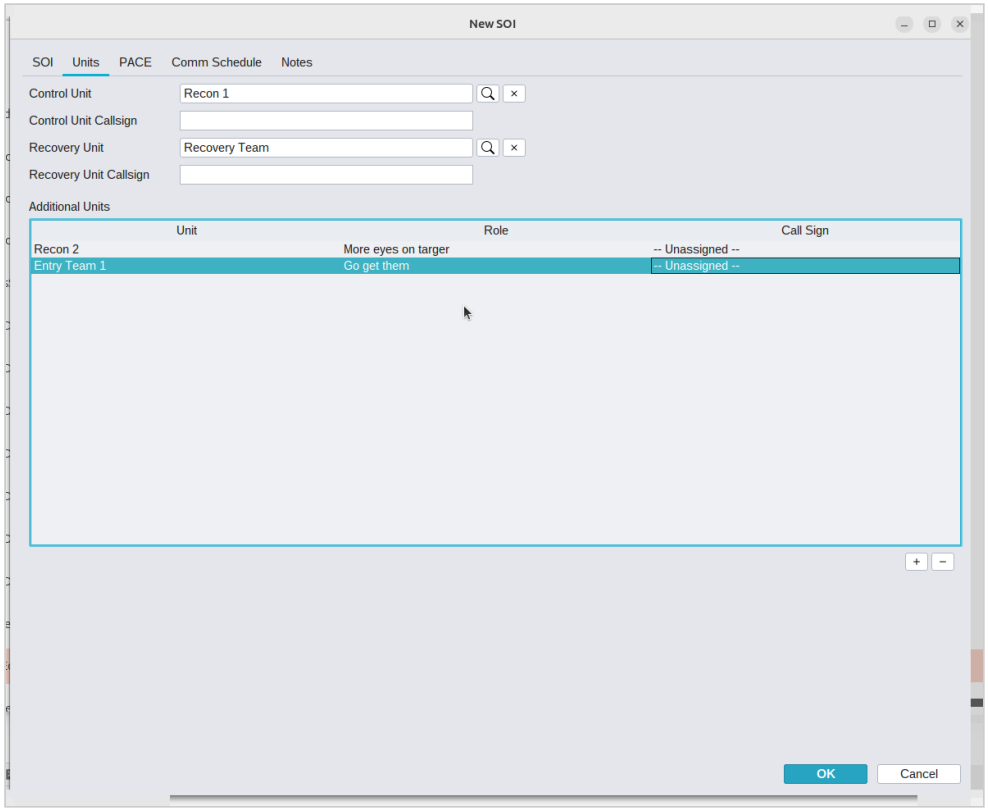
The SOI tab in the SOI Edit Dialog provides fields for the core details of the Signal Operating Instructions.

Fields

Field	Description
Title	The name or title of this SOI (required).
Objective	A description of the mission objective for this SOI.
Duration	The number of days this SOI covers (1–10). This determines the number of days available in the Comm Schedule.
Location	The location associated with this SOI. Click the button to edit the location, the search icon to select a different location, or the X icon to clear it.
Duress Code	A code word or phrase used to indicate communication under duress.
SARNEG	The SARNEG (Search and Rescue Numeric Encryption Grid) assigned to this SOI. A SARNEG is automatically generated if none exists.

> The Title field is required. If left blank, a validation error will be displayed when you click OK.

SOI Units



The Units tab in the SOI Edit Dialog assigns units to roles within this SOI.

Control and Recovery Units

Field	Description
Control Unit	The unit controlling this operation. Click the button to edit the unit, the search icon to select a unit, or the X icon to clear.
Control Unit Callsign	The callsign for the control unit within this SOI context.
Recovery Unit	The unit responsible for recovery operations. Click the button to edit, search to select, or X to clear.
Recovery Unit Callsign	The callsign for the recovery unit within this SOI context.

Additional Units

The Additional Units table lists other units participating in this SOI. Each unit can be assigned a role and callsign.

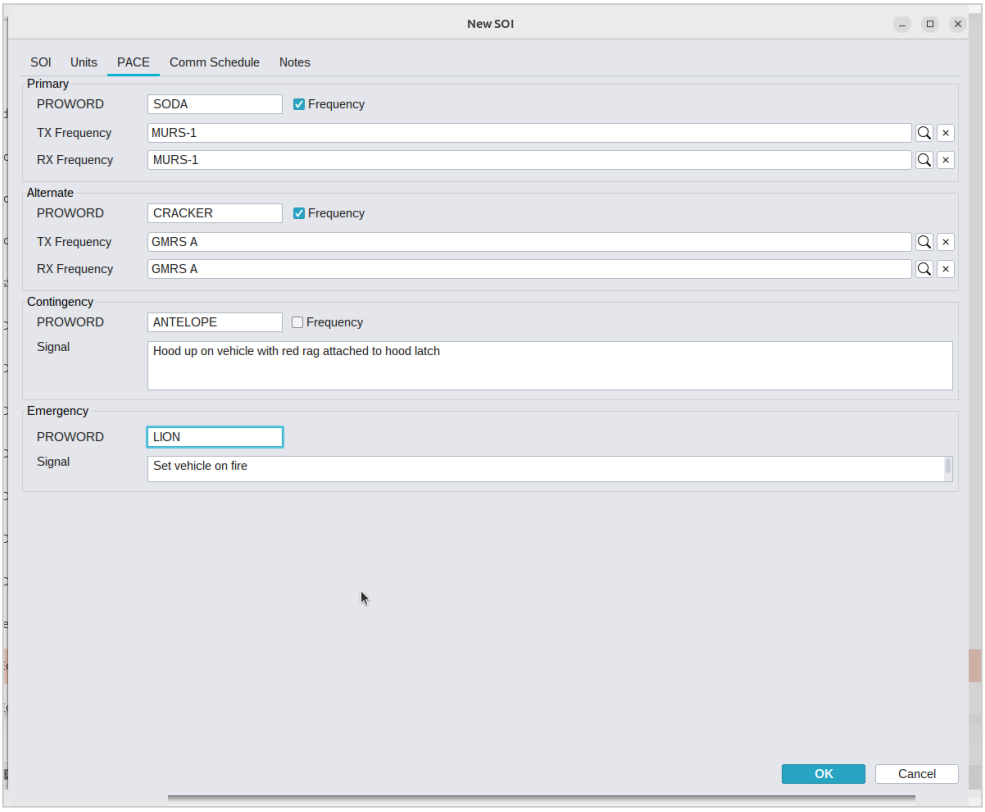
Column	Description

Unit	The unit name (read-only). Double-click a row to edit the unit.
Role	The unit's role in this SOI (editable inline).
Call Sign	The unit's callsign for this SOI (editable inline).

Use the **+** button to add units and the **−** button to remove selected units.

> Each unit can only appear once in an SOI — units already assigned as control, recovery, or additional units are excluded from the selection list.

SOI PACE Plan



The PACE tab in the SOI Edit Dialog configures the communications plan using the PACE (Primary, Alternate, Contingency, Emergency) framework. Each level defines how communication is established if the previous level fails.

PACE Levels

Each PACE level (Primary, Alternate, Contingency, Emergency) has the following fields:

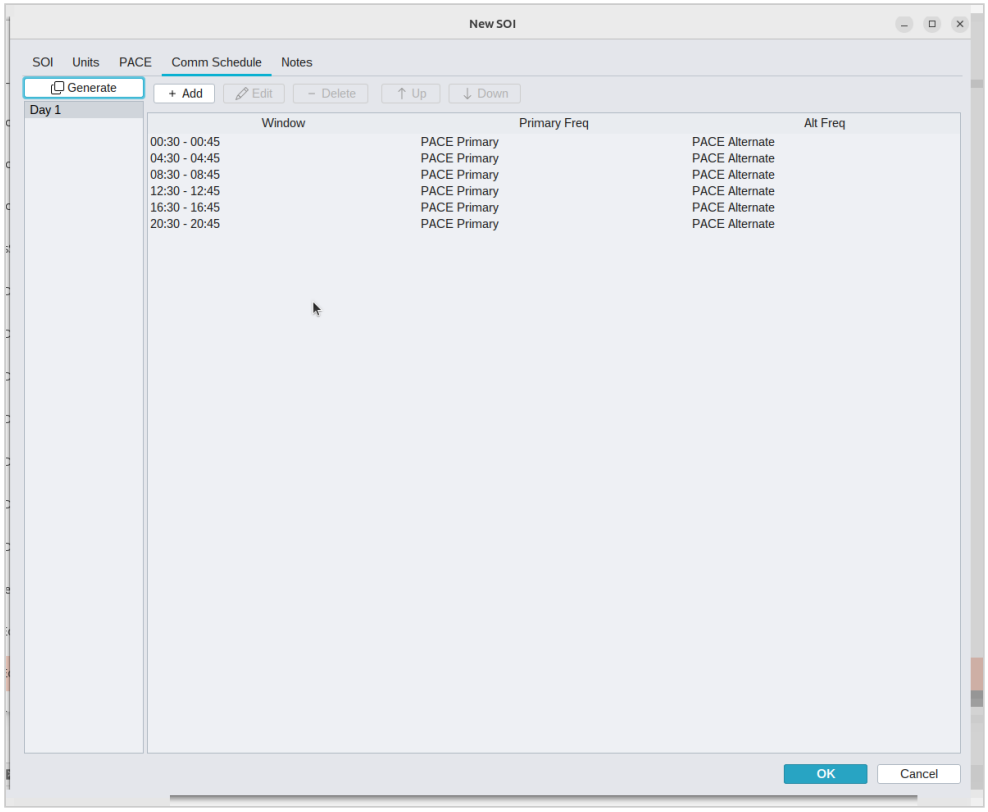
Field	Description
PROWORD	The procedural word used to initiate or identify this communication method.
Frequency (checkbox)	When checked, this level uses radio frequencies (TX and RX). When unchecked, a free-text signal description is used instead.
TX Frequency	The transmit frequency for this PACE level. Click to edit, search to select, or X to clear.
RX Frequency	The receive frequency for this PACE level. Can be the same as TX for simplex operations.
Signal	When the Frequency checkbox is unchecked, a free-text description of the signal method (e.g., visual signal, messenger).

Emergency Level

The Emergency level is always signal-based (no frequency option) and only has PROWORD and Signal fields.

> Frequencies assigned to one PACE level (e.g., Primary) are excluded from the selection list for other levels (e.g., Alternate, Contingency) to prevent duplication. TX and RX within the same level can share a frequency.

SOI Comm Schedule



The Comm Schedule tab in the SOI Edit Dialog defines the communication windows for each day of the SOI. The number of days is determined by the Duration field on the SOI Details tab.

Day List

The left panel shows the list of days. Select a day to view and edit its communication windows in the table on the right. Click **Generate** to use the Comm Schedule Generator to automatically create windows.

Communication Windows

The table on the right lists the communication windows for the selected day. Each window defines a time period and associated frequencies.

Actions

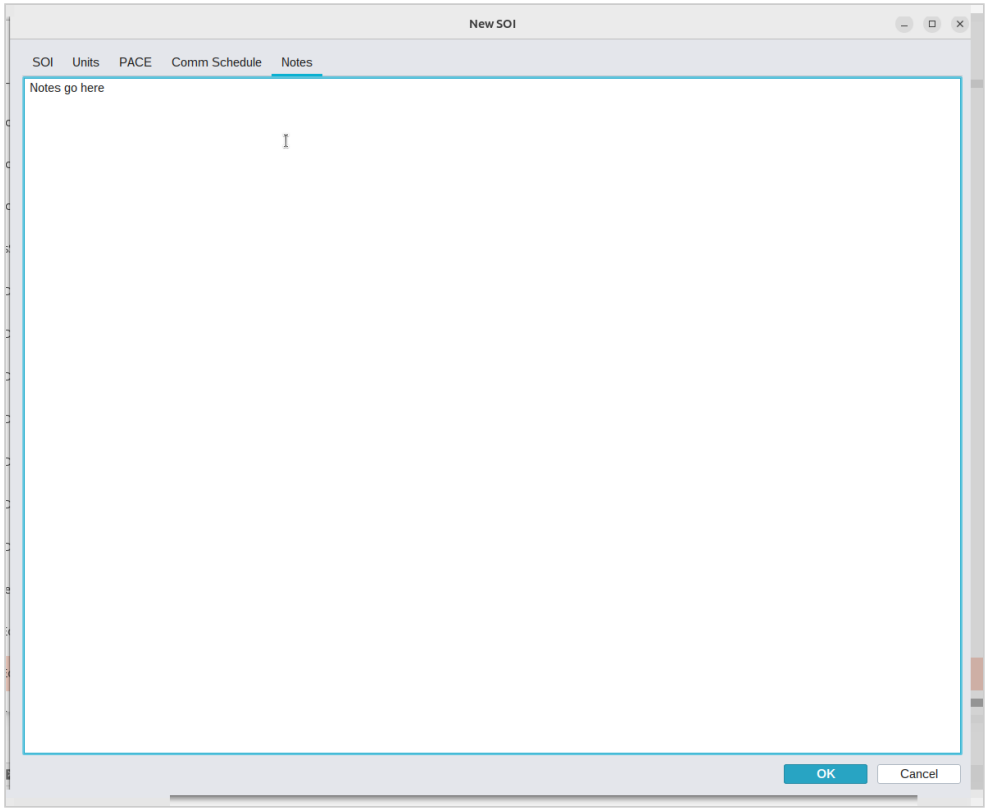
Button	Description
Add	Add a new communication window to the selected day. Opens the Comm Window Edit Dialog.
Edit	Edit the selected communication window. You can also double-click a row to edit.
Delete	Remove the selected communication windows from the schedule.

Up / Down	Reorder the selected window within the day's schedule.

Keyboard Shortcuts

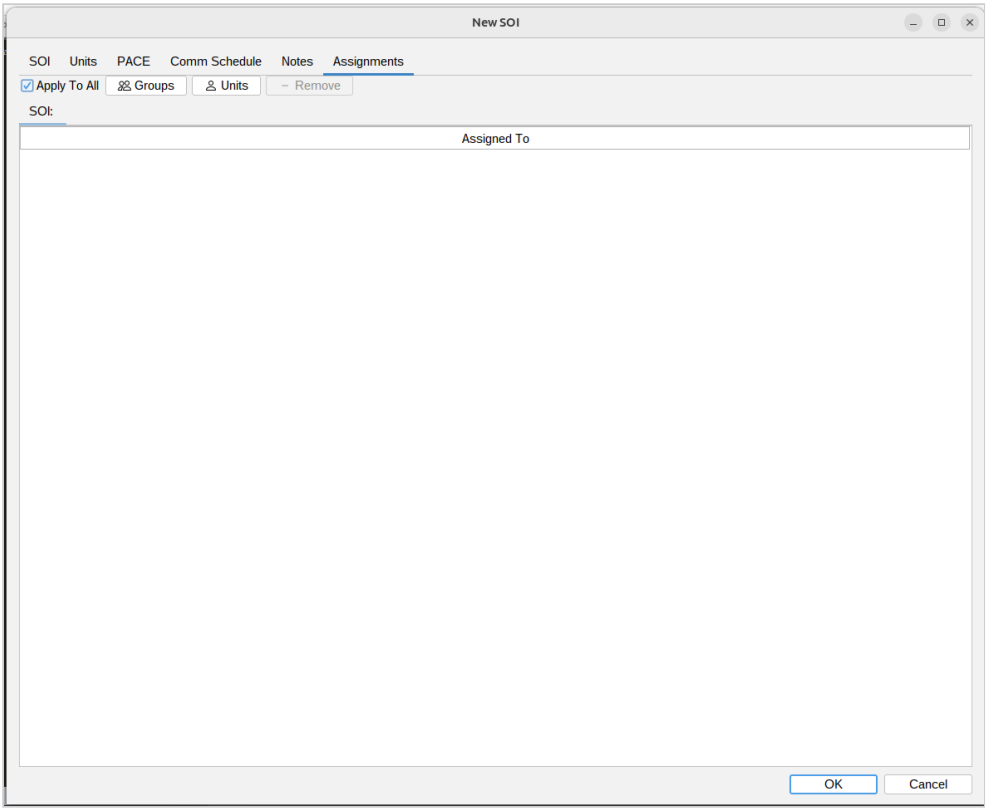
Key	Action
`Insert`	Add a new communication window.
`Delete`	Delete the selected communication window.
`Enter`	Edit the selected communication window.

SOI Notes



The Notes tab in the SOI Edit Dialog provides a free-text area where you can record notes about the SOI. Notes are saved with the SOI entity and included in the CEOI file.

SOI Assignments



An SOI is itself an asset. The Assignments tab in the SOI Edit Dialog displays and manages the units and groups this SOI is issued to.

> For the assets issued *to* this SOI -- DRYADs, SARNEGs, OTPs and Challenge/Response tables -- see the Assets tab instead.

This tab uses the Asset Assignment Panel to show current assignments and allow modifications.

Related Topics

- Asset Assignment Dialog
- SOI Assets
- SOI Panel

SOI Assets Tab

The Assets tab in the SOI Edit Dialog lists the cryptographic and communications assets that have been issued *to* this SOI -- DRYADs, SARNEGs, One-Time Pads, and Challenge/Response tables. It is the same view the Unit Assets Tab gives for a unit.

> Do not confuse this tab with the Assignments tab. The Assets tab shows what this SOI **holds**. The Assignments tab shows the units and groups this SOI **is issued to**, because an SOI is itself an asset.

Table Columns

Column	Description
Asset Type	The category of the asset: DRYAD, One Time Pad, SARNEG, or Challenge/Response. An SOI is never issued another SOI.
Id	The identifying name or label of the specific asset.
Print Template	The print template used for this asset when the SOI is printed. Click the cell to choose from the templates compatible with the asset type. If none is selected, the default template for that asset type is used.

Interactions

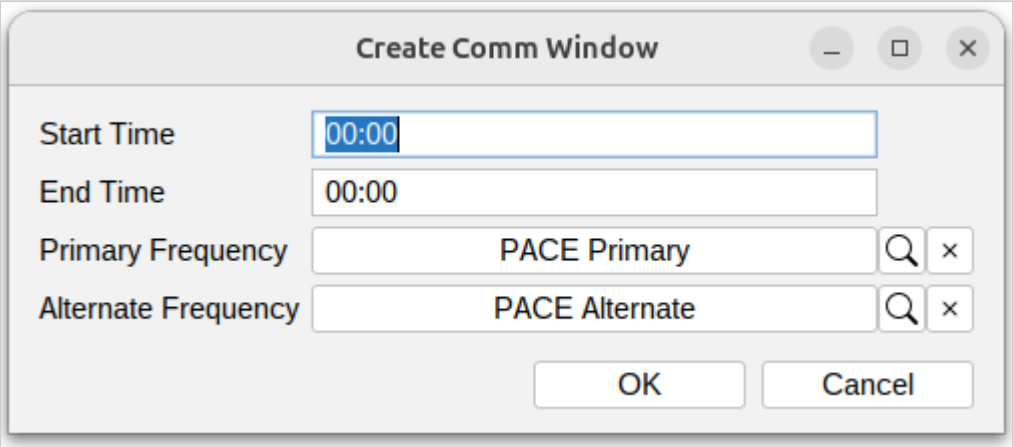
- **Double-click** -- Open a view dialog for that asset to inspect its full details.
- **Copy** -- Add a second instance of the selected asset, so it prints more than once in the SOI's packet.
- **Delete** -- Remove a row. Only copies can be removed; the original assignment is deleted from the asset's own editor.

> Which assets this SOI holds is decided from the asset's own editor -- from the DRYAD, OTP, SARNEG, or Challenge/Response panels in the main CEOI editor, using the Asset Assignment Panel and its SOIs button. This tab shows the result and controls how each asset prints.

Related Topics

- SOI Edit Dialog
- SOI Assignments
- Unit Assets Tab
- Asset Assignment Panel

Comm Window Edit



The Comm Window Edit dialog lets you create or edit a single communication window within an SOI communication schedule. A communication window defines a time range and the primary and alternate frequencies to use during that window.

Fields

Field	Description
Start Time	The start time of the communication window in 24-hour format (HH:MM).
End Time	The end time of the communication window in 24-hour format (HH:MM). Must be later than Start Time.
Primary Frequency	The primary frequency for this window. Click the button to view or edit the assigned frequency, or use the search icon to look up and assign a different frequency. Use the × button to reset to the SOI's default primary frequency.
Alternate Frequency	The alternate (backup) frequency for this window. Click the button to view or edit the assigned frequency, or use the search icon to look up a different frequency. Use the × button to reset to the SOI's default alternate frequency.

Frequency Buttons

Each frequency field has three controls:

- **Frequency button** — displays the currently assigned frequency name. Click to edit the frequency directly if one is already assigned.
- **Search icon (magnifying glass)** — opens a frequency selector to search for and assign a frequency.
- **Reset icon (×)** — resets the frequency back to the SOI's default primary or alternate frequency.

Saving Changes

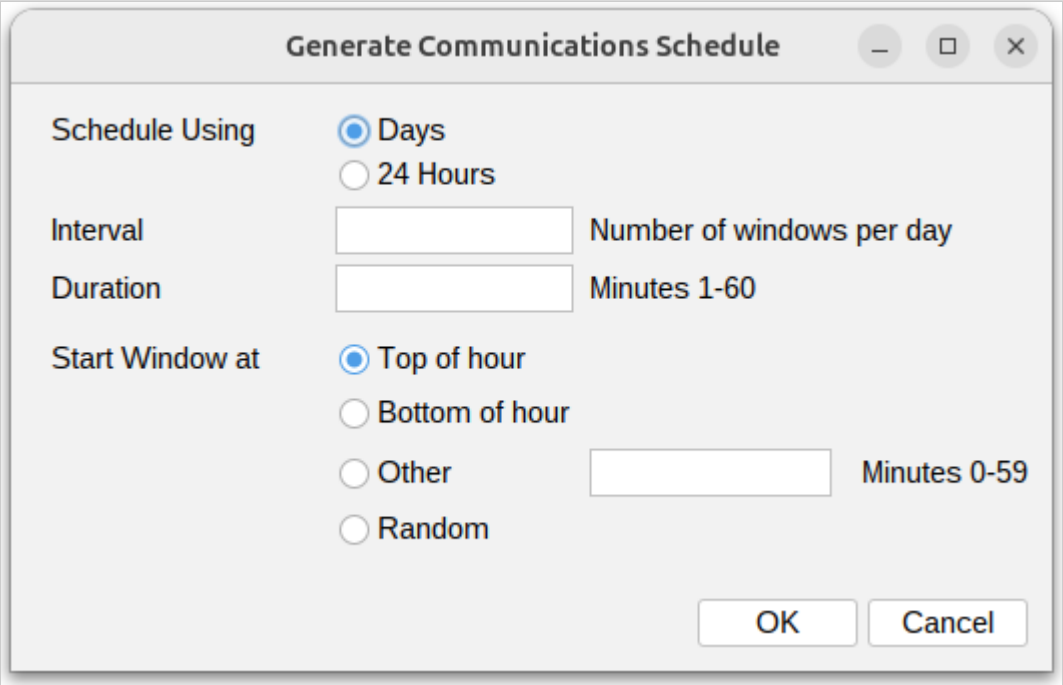
Click **OK** to validate and save the communication window. If the end time is not later than the start time, an error message is displayed and the dialog remains open.

Click **Cancel** to discard changes and close the dialog without saving.

Related Topics

- SOI Comm Schedule
- Comm Schedule Generator
- Frequency Edit Dialog

Comm Schedule Generator



The Comm Schedule Generator automatically creates a communication schedule for an SOI by calculating evenly-spaced communication windows across the SOI's duration. Use it as a quick starting point when building a new schedule from scratch.

> Generating a schedule will erase any existing communication windows already entered for the SOI. You will be asked to confirm before any existing data is overwritten.

Schedule Using

Choose how to express the interval between windows:

- **Times per Day** — specify how many communication windows occur each day. Scepter divides the day evenly to space the windows.
- **Hours** — specify an interval in hours and minutes (HH:MM). Windows are distributed across the entire SOI duration.

Interval

When scheduling *by times per day*, enter the number of windows per day (1–48).
When scheduling *by hours*, enter the interval in HH:MM format (e.g., `04:00` for every 4 hours).

Duration

Enter the length of each communication window in minutes (1–60).

Start Time

Choose where within each interval the window begins:

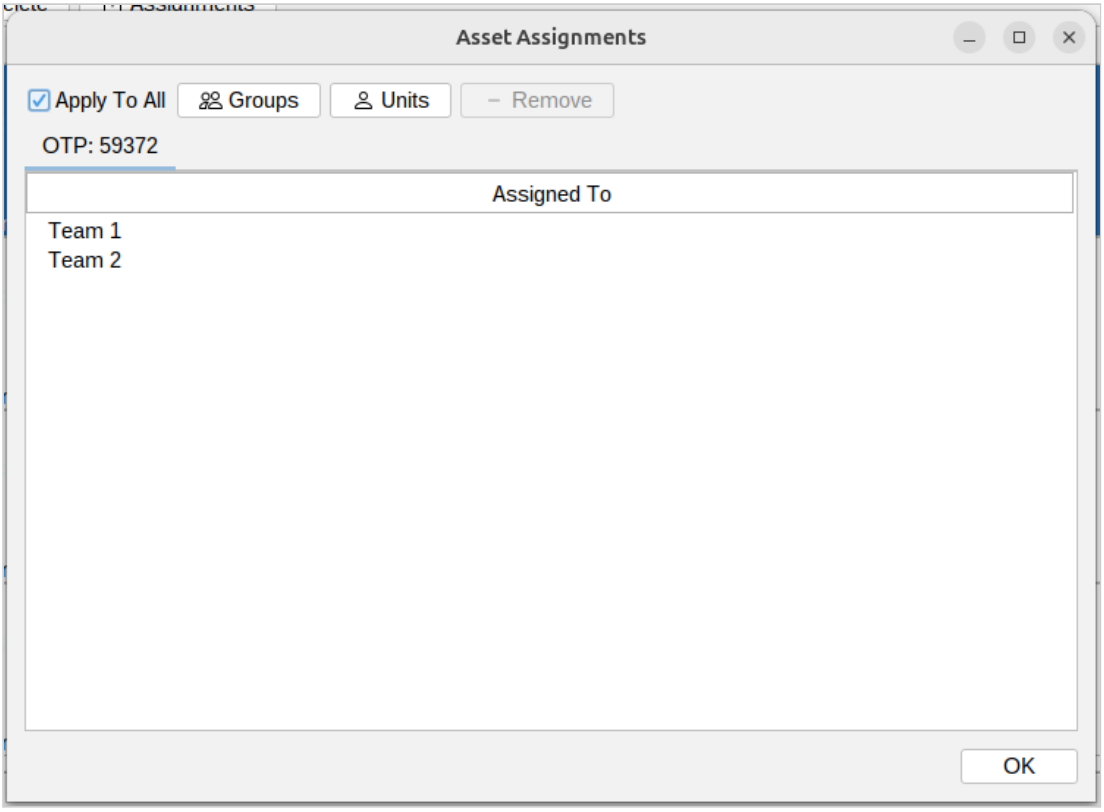
- **Top of Hour** — windows start at the top of each hour (e.g., 0000, 0400, 0800).
- **Bottom of Hour** — windows start 30 minutes into each hour (e.g., 0030, 0430, 0830).
- **Other** — specify a custom start offset in minutes (0–59) within the hour.
- **Random** — the start time is randomly offset by ±30 minutes from the calculated time, introducing variability.

> After generating a schedule you can fine-tune individual windows in the SOI Comm Schedule tab.

Related Topics

- SOI Comm Schedule
- SOI Edit Dialog
- Comm Window Edit Dialog

Asset Assignments



The Asset Assignments dialog manages which units, groups and SOIs are assigned to one or more assets (DRYADs, OTPs, SARNEGs, Challenge/Response tables, or SOIs). Open it from the toolbar or context menu of any asset panel in the main CEOI editor.

Layout

When multiple assets are selected, each asset appears on its own tab. Select a tab to view and manage assignments for that asset. The tab label shows the asset's name or ID.

Assignment Table

Each tab contains a tree-table listing the units, groups and SOIs currently assigned to that asset:

Column	Description
Assigned To	The name of the unit, group or SOI assigned to this asset. A row reading ****** ERROR ****** means the assignee has since been deleted.

Actions

Action	Description
Groups	Opens the Group Selector. Select one or more groups to assign the current asset to. The selected groups are added to the assignment list for the active tab's asset.

Units	Opens the Unit Selector. Select one or more units to assign the current asset to. The selected units are added to the assignment list.
SOIs	Opens the SOI Selector. Select one or more SOIs to issue the current asset with. The asset then appears on each SOI's Assets tab and prints as part of that SOI's packet. This button is hidden when the assets being assigned are themselves SOIs — an SOI is never issued another SOI.
Remove	Removes the selected rows from the assignment table. The button is only enabled when at least one row is selected.
OK	Closes the dialog. You can also press `Escape` to close.

Double-Click to Edit

Double-click any row in the assignment table to open the edit dialog for that unit, group or SOI, letting you inspect or modify the entity without leaving the assignments dialog.

> You can open this dialog with multiple assets selected at once — each asset appears on its own tab so you can manage several assignments in a single session.

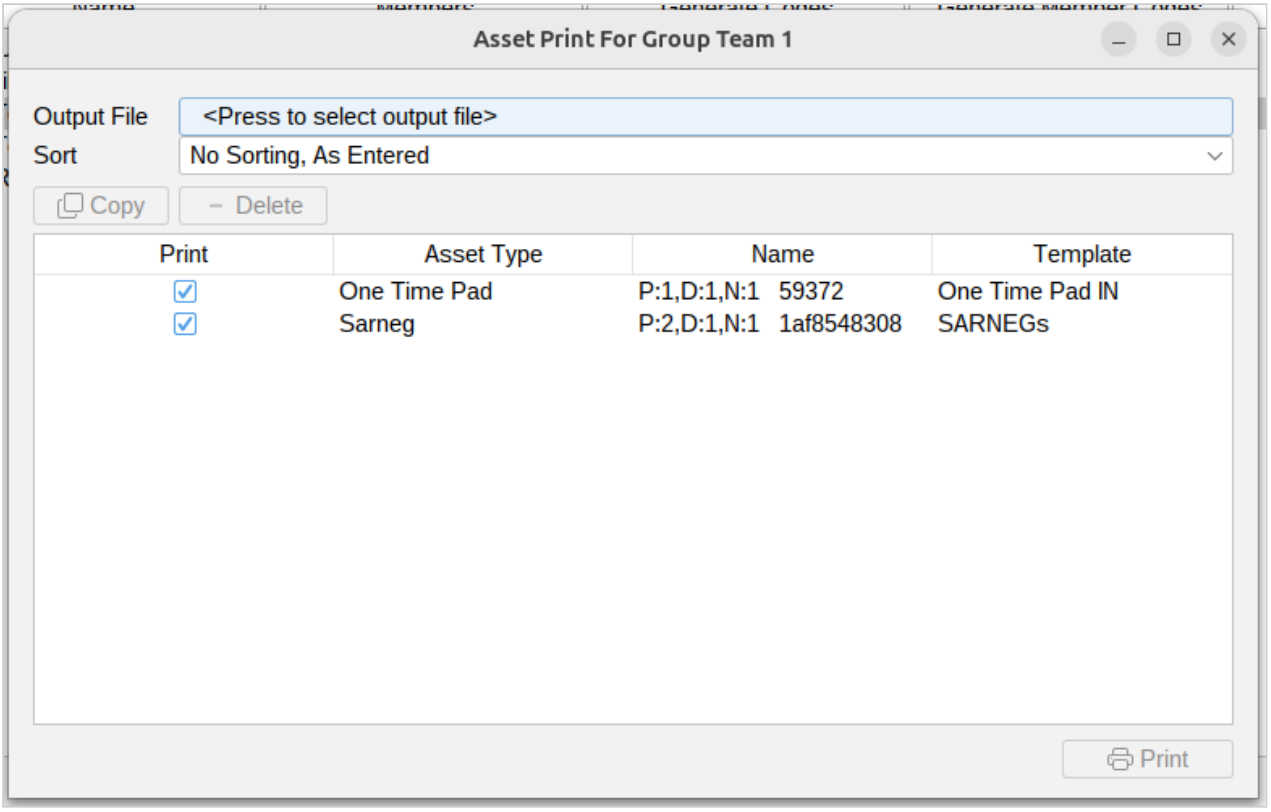
> Assigning an asset to a group gives every member of that group access to the asset.

> Assignments take effect immediately. Adding or removing a unit or group is applied right away — the **OK** button simply closes the dialog.

Related Topics

- Unit Selector
- Unit Assets Tab -- view assets assigned to a unit
- Group Assets Tab -- view assets assigned to a group
- SOI Assets Tab -- view assets issued with an SOI
- DRYADs Panel
- One-Time Pads Panel
- SARNEGs Panel

Asset Print



The Asset Print dialog lets you print the assets assigned to a unit, group, or SOI. It lists all assets (frequencies, DRYADs, SARNEGs, OTPs, challenge/response pairs, units, and sub-groups) belonging to the holder and lets you select which ones to print and which print template to use for each.

Output File

Click the **Output File** button to choose a file path where the printed output will be saved. The button label updates to show the selected file path once a file is chosen. The **Print** button remains disabled until a valid output file is selected.

Sort Order

Use the **Sort Order** dropdown to control the order in which assets appear in the printed output:

- **As Entered** — preserves the original assignment order.
- **Type / Period / Division / Number** — sorts first by asset type, then by period, division, and number.
- **Period / Division / Type / Number** — sorts first by period, then by division, type, and number.

Asset Table

Column	Description
Print	

	Checkbox to include or exclude the asset from the print job. All assets are selected by default.
Asset Type	The type of asset (e.g., Frequency, DRYAD, SARNEG, OTP, Unit, Group).
Name	The name of the asset. For the unit and group asset types a descriptive label — **Member Units** or **Member Groups** — is shown instead of a name.
Template	The print template to use for this asset. Click to select from a dropdown of templates available for the asset type.

Copy and Delete

Select one or more rows in the table and use the **Copy** or **Delete** toolbar buttons (or the right-click context menu) to duplicate or remove print items. Copying an item lets you print the same asset multiple times with different templates.

Printing

Click **Print** to generate output for all checked assets using their selected templates. Each asset is rendered using the template chosen in the Template column.

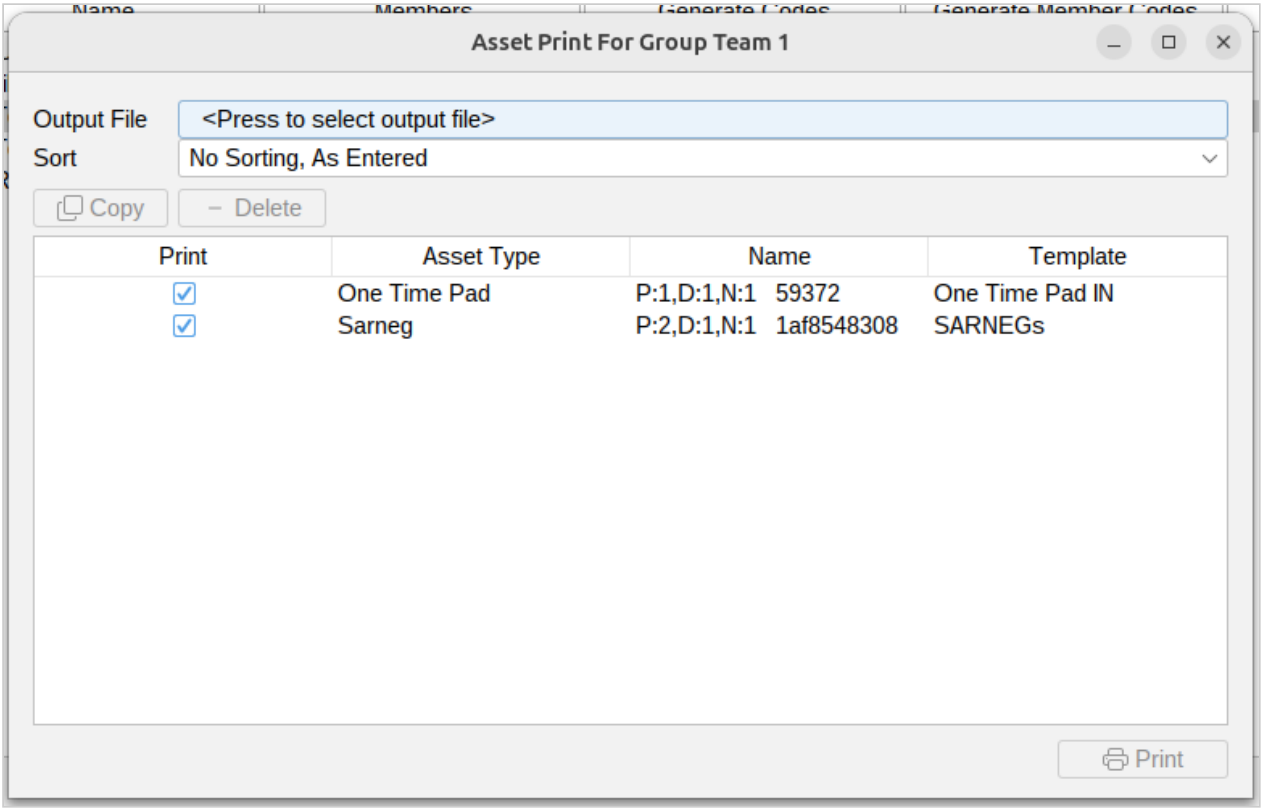
Every row is narrowed to its own asset, so a row prints that asset and no other of its type. The **Member Units** and **Member Groups** rows are the exception: they stand for every unit or group the holder covers rather than for one entity, so they are printed unnarrowed and their contents follow the holder itself.

> You can selectively print specific assets by unchecking the ones you do not need. This is useful when you only need to reprint a single asset type rather than all assets belonging to the unit or group.

Related Topics

- Unit Edit Dialog
- Unit Assets
- Group Edit Dialog
- Group Assets
- Asset Assignment Dialog
- Print Templates

Group Asset Print



The Group Asset Print dialog lets you print the assets assigned to a group. It lists all assets (frequencies, DRYADs, SARNEGs, OTPs, challenge/response pairs, units, and sub-groups) belonging to the group and lets you select which ones to print and which print template to use for each.

Output File

The **Output File** field at the top specifies the file path where the printed output will be saved. Click the folder icon to browse for a location.

Asset Table

Column	Description
Print	Checkbox to include or exclude the asset from the print job. All assets are selected by default.
Asset Type	The type of asset (e.g., Frequency, DRYAD, SARNEG, OTP, Unit, Group).
Name	The name of the asset. For unit and group asset types, a descriptive label is shown instead.
Template	The print template to use for this asset. Click to select from a dropdown of templates available for the asset type.

Printing

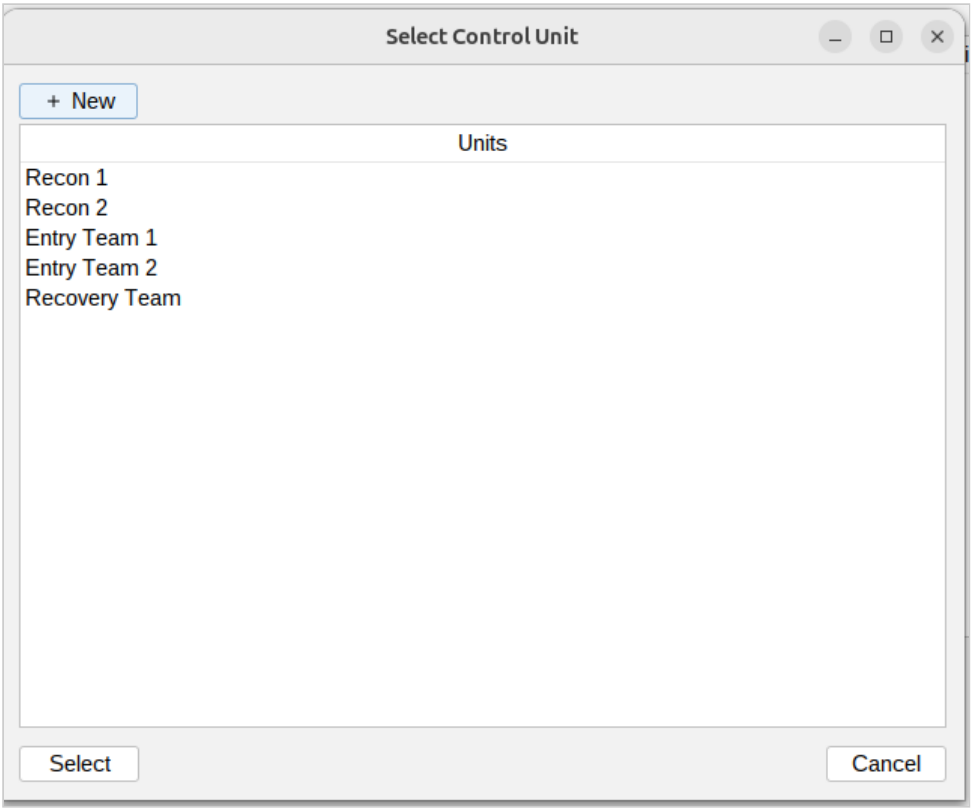
Click **Print** to generate output for all checked assets using their selected templates. Each asset is rendered using the template chosen in the Template column.

> You can selectively print specific assets by unchecking the ones you do not need. This is useful when you only need to reprint a single asset type rather than the entire group.

Related Topics

- Group Edit Dialog
- Group Assets
- Asset Assignment Dialog
- Print Templates

Entity Selector



The Entity Selector is a general-purpose selection dialog used throughout Scepter. It presents a list of entities -- such as units, groups, frequencies, locations, or other CEOI items -- and allows you to pick one or more from the list. The dialog title and column header change depending on the type of entity being selected.

Table

The dialog displays a single-column table listing all available entities of the requested type. Each row shows the entity's display name. Click a row to select it, or hold `Ctrl` (or `Cmd` on macOS) and click to select multiple rows.

Actions

Action	Description
New	Click the **New** button (with the **+** icon) to create a new entity of the appropriate type. The corresponding edit dialog opens. After you save the new entity, it is added to the list and automatically selected.
Select	Click **Select** to confirm your selection and return the chosen entities to the calling dialog.
Cancel	Click **Cancel** or press `Escape` to close the dialog without making a selection.

Quick Selection

You can double-click any row in the table to immediately select that single entity and close the dialog. This is equivalent to clicking the row and then clicking **Select**.

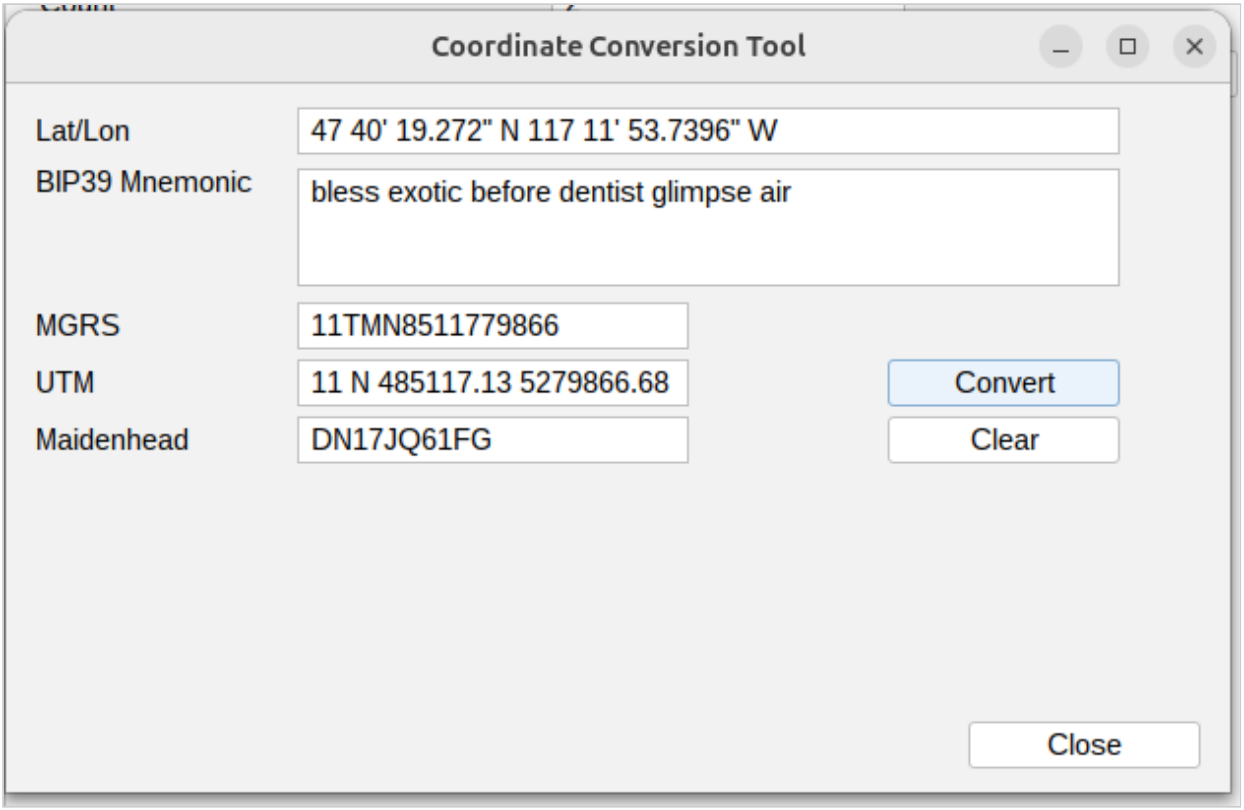
> When the selector is opened in single-selection mode, you can only pick one entity at a time. In multi-selection mode, hold `Ctrl` or `Shift` while clicking to select multiple items before confirming with the **Select** button.

> The Entity Selector is used as the underlying selection mechanism for many operations in Scepter, including adding units to groups, selecting frequencies, and choosing locations. The specific appearance and available entities depend on the context in which the selector is opened.

Related Topics

- Unit Selector -- specialized selector for units
- Unit Edit Dialog
- Group Edit Dialog

Coordinate Conversion Tool



The Coordinate Conversion Tool converts a geographic coordinate between multiple formats simultaneously. Enter a coordinate in any one field and click **Convert** to populate all remaining fields.

Coordinate Formats

Field	Format	Example
Lat / Lon	Decimal degrees or degrees/minutes/seconds in either hemisphere	38.8977 -77.0366
BIP-39	Multi-word mnemonic phrase encoding a latitude/longitude pair	(word phrase)
MGRS	Military Grid Reference System	18S UJ 23428 07396
UTM	Universal Transverse Mercator	18N 323428 4307396
Maidenhead	Amateur radio grid locator (Maidenhead Locator System)	FM18lv

Converting a Coordinate

1. Enter a coordinate in any one of the fields above.
2. Click **Convert**. All other fields are populated with the equivalent coordinate in each format.

Clearing Fields

Click **Clear** to erase all fields and start a fresh conversion.

> Only enter a value in one field at a time. The tool uses whichever field contains text as the input source, checking in order: Lat/Lon, BIP-39, MGRS, UTM, Maidenhead.

Related Topics

- Location Edit Dialog
- CEOI Locations

QR Code Generator



The QR Code Generator converts text into a scannable QR code image. It can be opened manually from the Tools menu or automatically by other dialogs in Scepter (such as the DRYAD Viewer and OTP dialogs) to share data with mobile devices.

Text Tab

Enter or paste the text you want to encode as a QR code into the text area. Click the **QR Code** tab to generate and view the code.

QR Code Tab

When you switch to the **QR Code** tab the QR code image is generated automatically from whatever text is in the Text tab. The image updates each time you switch to this tab.

Save as Image

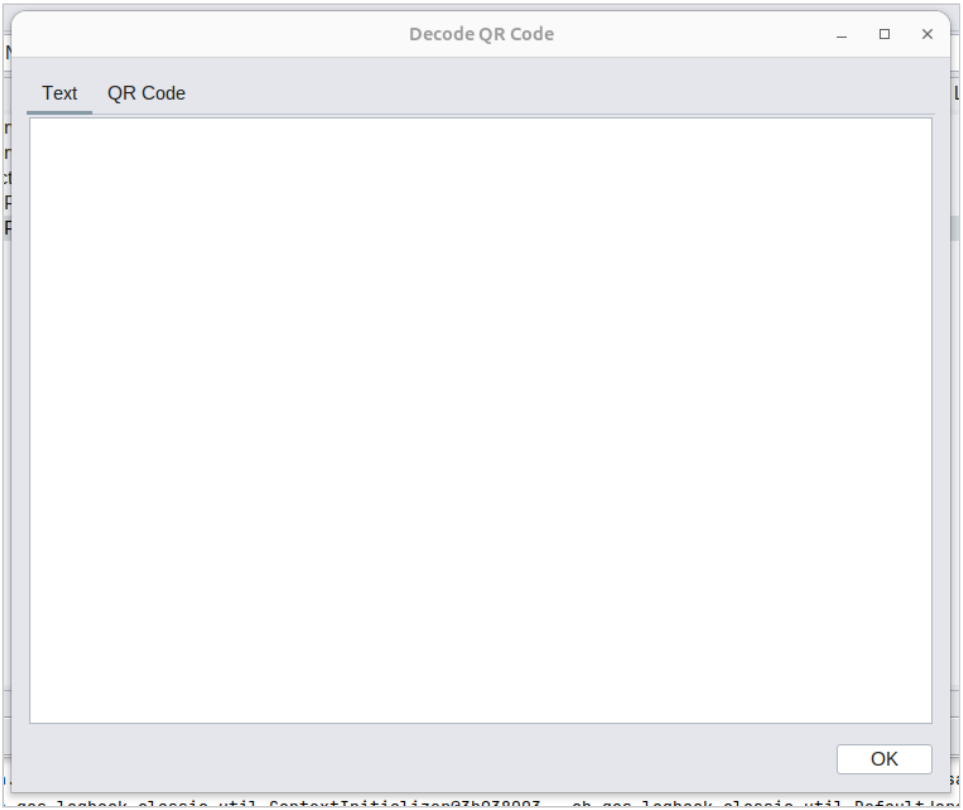
Click **Save as Image** to export the QR code as a PNG file. A file chooser dialog opens so you can pick the save location. The `.png` extension is added automatically if not already present.

> QR codes can hold a limited amount of data. Very long text (such as a large OTP) may produce a dense QR code that is harder to scan. Consider splitting the data across multiple codes if scanning fails.

Related Topics

- QR Code Decoder
- DRYAD Viewer
- OTP Import

QR Code Decoder



The QR Code Decoder reads a QR code using your computer's camera or a captured image and extracts the encoded text. It is used by other Scepter dialogs (such as OTP Import) to scan QR codes and return their content automatically.

Image Tab

The **Image** tab shows the camera capture panel. Point your camera at a QR code; the panel will automatically detect and decode it.

- When the QR code is decoded, the text content is extracted and control returns to the calling dialog.
- In standalone mode (opened from the Tools menu), the decoded text is shown on the **Text** tab.

Text Tab

After a successful decode, the decoded text appears in the read-only text area on the **Text** tab. You can select and copy the text from here.

Closing

Click **OK** to close the dialog.

> Ensure good lighting and hold the camera steady for reliable QR code detection. If the code is small, try moving the camera slightly closer.

Related Topics

- QR Code Generator
- OTP Import Dialog
- Camera Capture Dialog

Camera Capture

The Camera Capture dialog uses your computer's camera to scan a QR code and extract the encoded text. It is invoked automatically by other Scepter dialogs (such as the QR Code Decoder and OTP Import) when they need to scan a QR code.

How It Works

1. When the dialog opens, the camera activates and displays a live preview.
2. Hold a QR code up to the camera.
3. When Scepter detects and decodes the QR code, the camera beeps, the preview stops, and the dialog closes automatically, returning the decoded text to the calling dialog.

No Camera Found

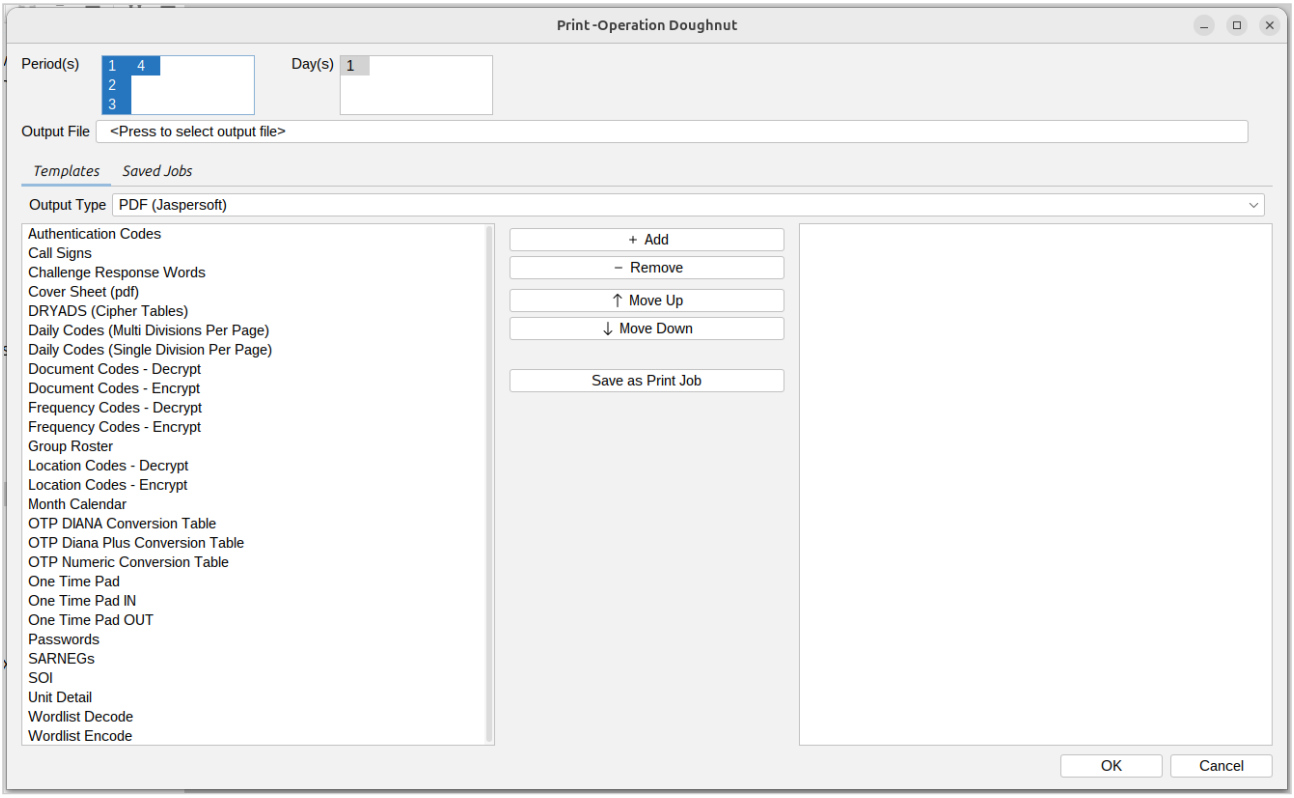
If no camera is detected, the dialog displays a message instead of a live preview. In this case, close the dialog and use the file-based QR code decoder option if available, or use a mobile device to decode the QR code manually.

> Ensure adequate lighting and hold the QR code steady and fully within the camera's field of view. If scanning fails, try moving the code closer to the camera.

Related Topics

- QR Code Decoder
- OTP Import

Print Job



The Print Job dialog lets you define and run a print job — selecting which templates to print, the periods and divisions to include, and the output file. You can build a custom set of templates on the fly or choose a previously saved job configuration.

Output File

Click the **Output File** button to choose the destination file for the printed output. A print job cannot be started until an output file is selected.

Period and Division Filters

Use the **Period** and **Division** lists to select which periods and divisions to include in the output. All periods and divisions are selected by default. Hold Ctrl or Shift to select multiple entries.

Templates Tab

Use the Templates tab to build a custom print job by assembling a list of templates to run:

- **Output Type** — filter the available templates list by type (e.g., PDF, HTML).
- **Available Templates** — the left list shows templates available for the selected type. Double-click or click **Add** to move them to the selected list.
- **Selected Templates** — the right list shows templates that will be included in the print job, in order. Double-click or click **Remove** to remove a template.
- **Move Up / Move Down** — reorder templates within the selected list.

- **Save as Job** — saves the current template selection as a named job configuration for reuse later.

Saved Jobs Tab

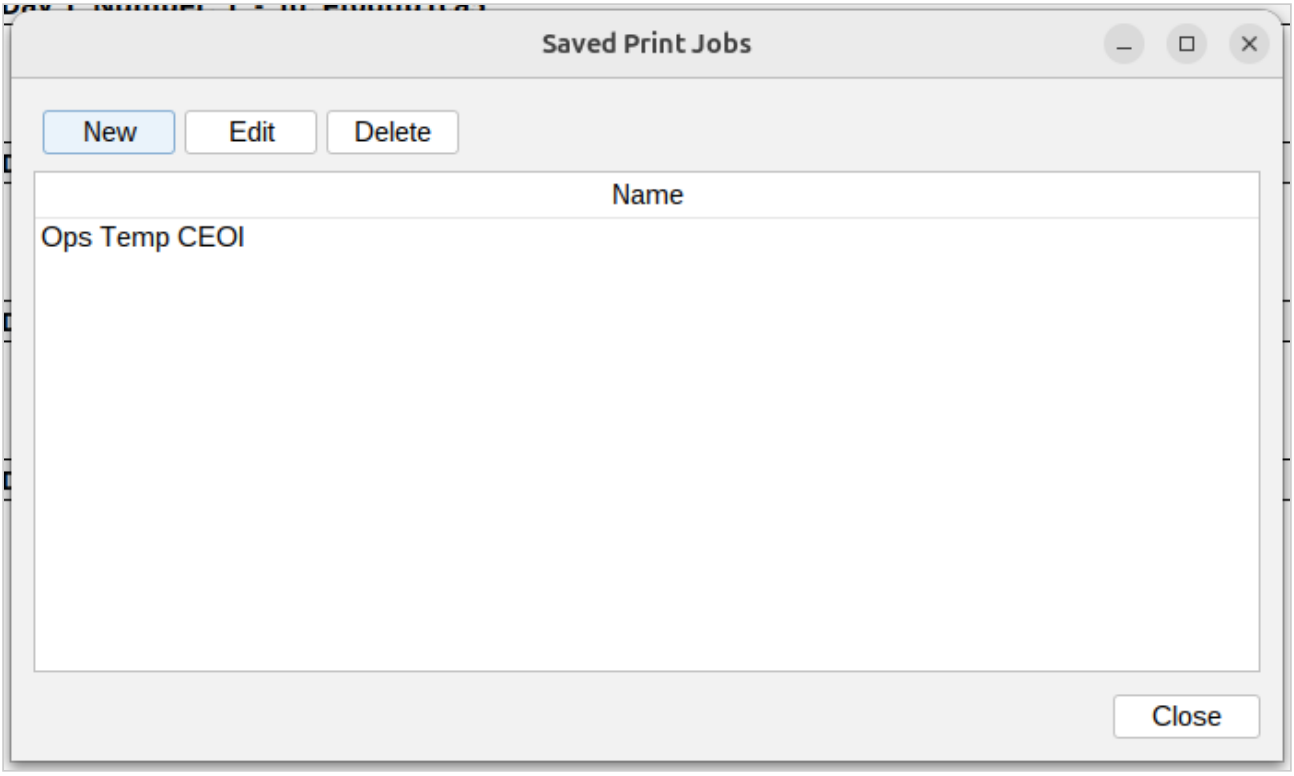
The Saved Jobs tab shows all previously saved print job configurations. Select a row to load it; the Templates tab will update to reflect the saved job's template list.

> Save frequently-used template combinations as named jobs to avoid rebuilding them each time you print.

Related Topics

- Print Configurations
- Print Templates
- Print Configuration Editor

Print Configurations



The Print Configurations dialog manages saved print configurations. A print configuration is a named, reusable collection of print templates in a specific order and output type. Configurations appear in the Saved Jobs tab of the Print Job dialog.

Table Columns

Column	Description
Name	The configuration name.
Type	The output type (e.g., PDF, HTML).
Items	The number of print templates included in the configuration.

Actions

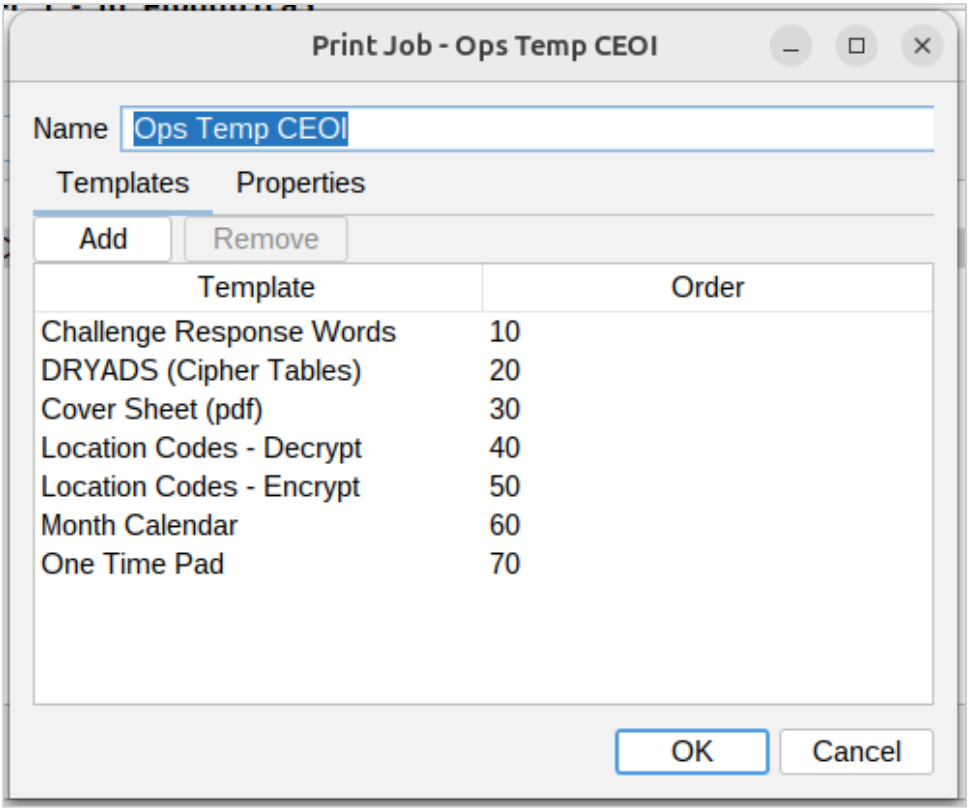
- **New** — opens the Print Configuration Editor to create a new configuration.
- **Edit** — opens the selected configuration in the Print Configuration Editor. Double-clicking a row also opens it for editing.
- **Delete** — permanently removes the selected configuration.

Related Topics

- Print Configuration Editor
- Print Job Dialog

- Print Templates

Print Configuration Editor



The Print Configuration Editor creates or edits a named print configuration — a saved, reusable ordered set of print templates. Configurations appear in the Saved Jobs list of the Print Job dialog.

Name

Enter a descriptive name for this configuration. The name must be unique among all saved configurations.

Templates Tab

The Templates tab manages the list of print templates included in this configuration.

- **Add** — opens the Print Templates selection dialog to pick a template and add it to the configuration.
- **Remove** — removes the selected template from the configuration. Templates are sorted by their assigned order value.

The table lists each included template along with its order number. Templates are printed in ascending order.

Properties Tab

The Properties tab contains a free-form text area for configuration-level properties. These properties are passed to the print engine and may influence print output formatting. Property keys and values use the format `key=value`, one per line.

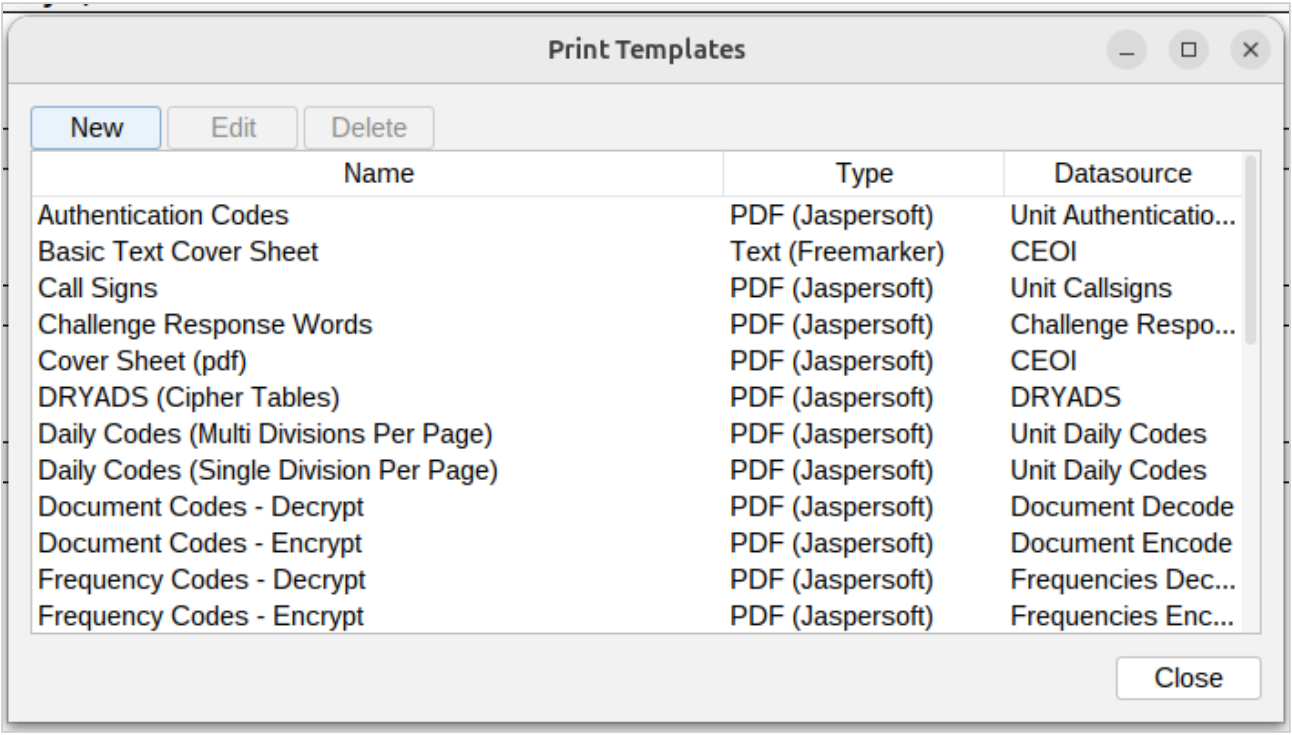
Saving

Click **OK** to save the configuration and return to the Print Configurations dialog. Click **Cancel** or press **Escape** to discard all changes.

Related Topics

- Print Configurations
- Print Templates
- Print Job Dialog

Print Templates



The Print Templates dialog manages the library of print templates available to Scepter. Each template defines how a specific type of CEOI content is formatted for output (e.g., a unit's codes, a frequency list, or an OTP page). Templates can be created, edited, and deleted here.

Table Columns

Column	Description
Name	The template name, as it appears in print job and asset print dialogs.
Type	The output type (e.g., PDF, HTML).
Entity Type	The type of CEOI content the template is designed to print.

Actions

- **New** — opens the template editor to create a new print template.
- **Edit** — opens the selected template in the editor. Double-clicking a row also opens it for editing.
- **Delete** — permanently removes the selected template.

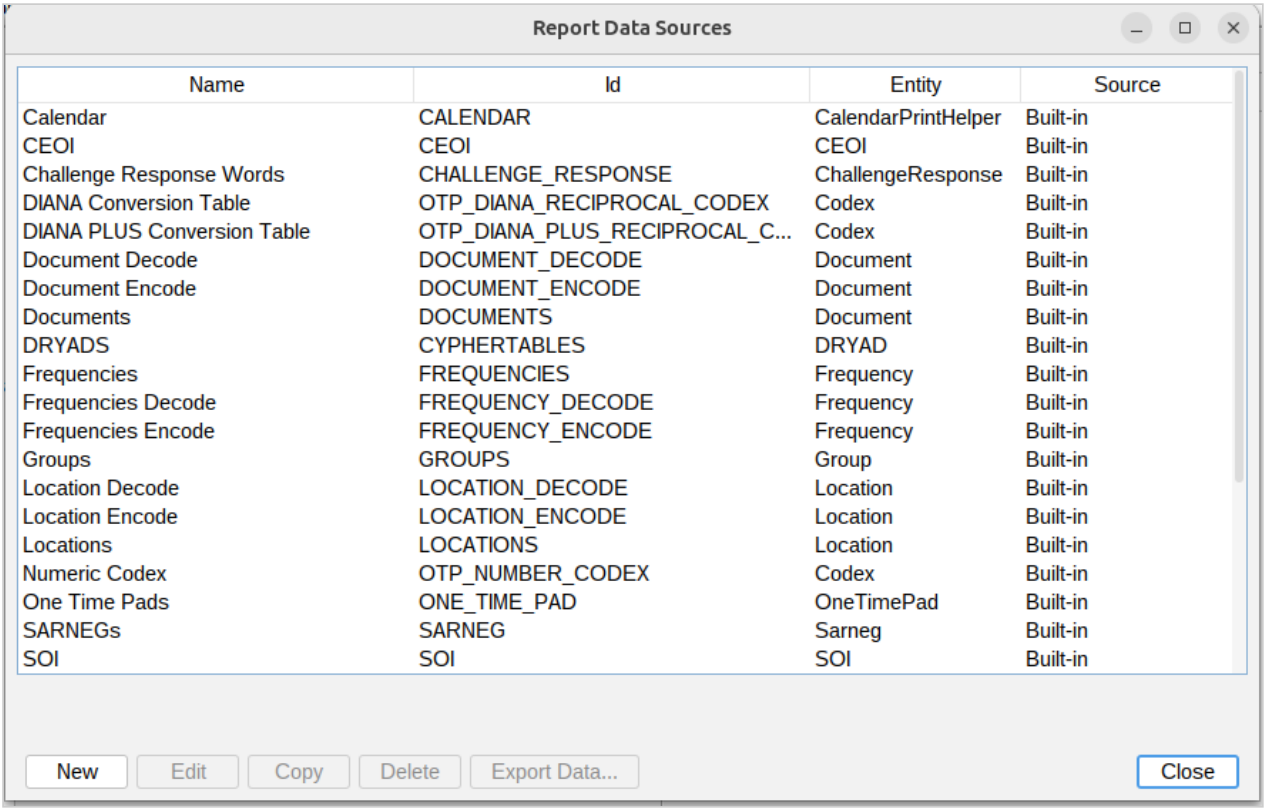
Selection Mode

When the dialog is opened in selection mode (from within another dialog), a **Select** button appears. Click **Select** to return the chosen template to the calling dialog.

Related Topics

- Print Job Dialog
- Print Configurations
- Print Configuration Editor
- Asset Print Dialog

Report Data Sources



A print template decides how a report *looks*. A data source decides what data it is given. This dialog lists every data source available to Scepter: the ones that ship with it, and any you have written yourself.

Each data source is a small JavaScript file that reads the open CEOI and returns rows. Because they are files rather than compiled code, you can read any of them, copy one as a starting point, and write your own without rebuilding Scepter.

Table Columns

Column	Description
Name	The name shown in the print template editor's data source list.
Id	The stable identifier saved inside a template. Renaming a data source does not break templates; changing its id does.
Entity	The kind of CEOI content it produces, which decides where it appears for right-click printing.
Source	**Built-in** ships with Scepter and is replaced on update. **Mine** is yours, in `config/datasources`, and is never touched by an update.

Actions

- **New** — opens the editor on a working starter script you can run immediately.
- **Edit** / **View** — opens the selected script. Your own scripts open for editing; built-in ones open read-only, because an update would overwrite any change.
- **Copy** — starts a new script of your own from the selected one. This is the way to base a data source on a built-in.
- **Delete** — removes one of your scripts. Templates pointing at it will no longer print, so check first.
- **Export Data...** — runs the data source against the open CEOI and writes every row to a JSON file. See below.

Designing a template against real data

Export Data... is the bridge between Scepter and Jasper Studio. It writes the same JSON that Scepter feeds to the report engine when it prints, so a template designed against the exported file fills correctly the first time you run it inside Scepter.

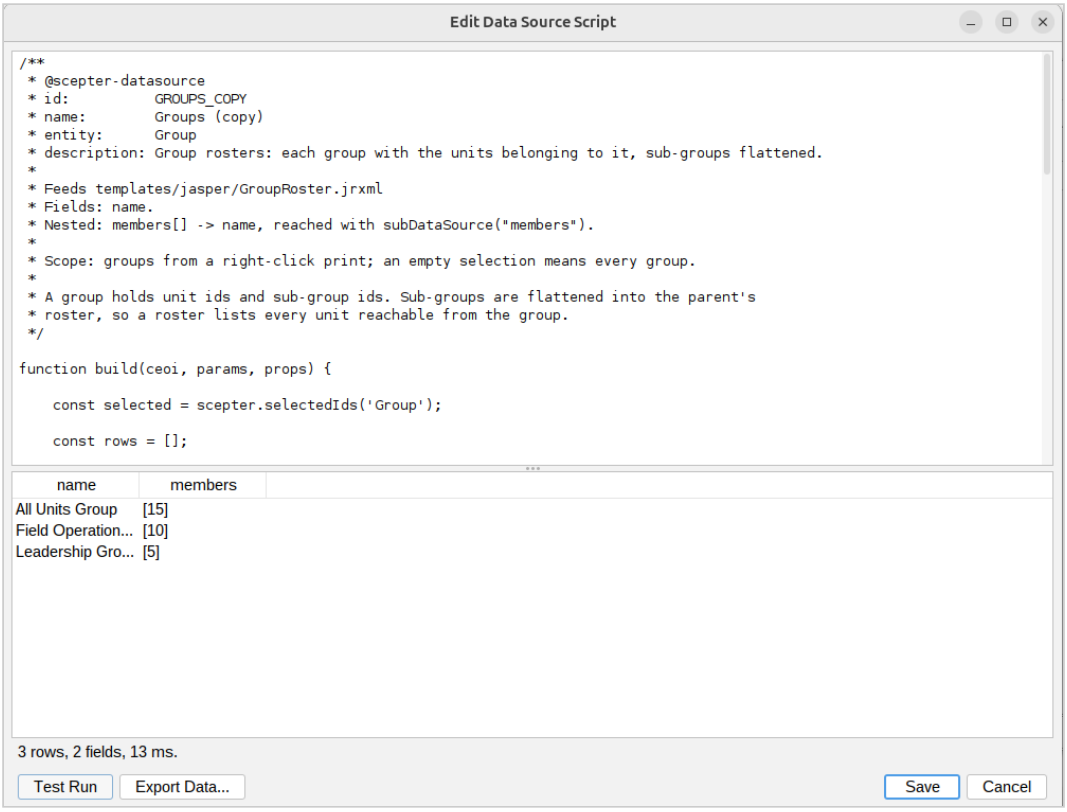
1. Select a data source and click **Export Data...**.
2. In Jasper Studio, create a *JSON* data adapter pointing at the file you saved.
3. Design the template. The fields are discovered from the file, and the preview shows your own data rather than sample values.
4. Save the `.jrxml` into your templates directory and register it in Print Templates, pointing it at this same data source.

Nested data — a group's members, an SOI's PACE plan — appears in the file as an array inside each row. In a template these are reached with a subreport or list component whose data source expression is `((net.sf.jasperreports.json.data.JsonQLDataSource) $P{REPORT_DATA_SOURCE}).subDataSource("members")`.

Handle exported files carefully

An exported file contains live CEOI data — call signs, authentication codes, passwords, whatever the data source produces — and it is written unencrypted. Treat it exactly as you would a printed CEOI, and delete it once the template is finished.

Data Source Script Editor



A data source script reads the open CEOI and returns the rows a report prints. This editor lets you write one, run it against your real data, and see the rows before any template exists.

The shape of a script

Every script has a header naming it, and a `build` function returning an array of row objects. The keys of each row are the field names a template refers to as `$F{name}`.

```
/**
 * @scepter-datasource
 * id:          MY_ROSTER
 * name:        My Roster
 * entity:      Unit
 * description: What this data source produces.
 */

function build(ceoi, params, props) {
    const rows = [];
    for (const unit of ceoi.getUnits())
        rows.push({ name: unit.getName() });
    return rows;
}
```

The **id** is what a template saves, so it must be unique and should not change once templates point at it. The **entity** decides which right-click print menus the report appears in.

Scope — periods, divisions and selections

A script does not decide what to include on its own. The print dialog's period and division lists arrive as `scepter.selectedPeriods()` and `scepter.selectedDivisions()`, and a right-click print on a table row arrives as `scepter.selectedIds('Unit')`, where an empty selection means all. Honoring these is what makes a report obey the print dialog.

Test Run

Test Run executes the script against the open CEOI, over every period and division with nothing selected, and shows the rows in the table below the editor. The status line reports how many rows and fields came back and how long it took. A script error is reported with its line number rather than as a failed print job.

Nested arrays show in the table as a count, such as `[4]`, because they are a subreport's data rather than a column. Export the data to see inside them.

Export Data...

Writes *every* row — not just the ones shown in the table — as JSON, for opening in Jasper Studio's JSON data adapter. It is the same JSON Scepter feeds the report engine when it prints, so a template designed against the file fills correctly the first time it runs inside Scepter.

The file contains live CEOI data and is written unencrypted. Treat it as you would a printed CEOI, and delete it when the template is finished.

Saving

Save checks the script first: that it parses, that it has a header with an id, that the id is not already taken, and that it defines a `build` function. Anything wrong is reported before the file is written, and the cursor moves to the offending line. Scripts are saved into `config/datasources` and are available immediately, without restarting Scepter.

Built-in scripts open read-only, because an update replaces them. Use **Copy** in Report Data Sources to base your own on one.

Writing Custom Data Sources

A data source is a JavaScript file that reads the open CEOI and returns the rows a report prints. Writing one lets you produce a report Scepter does not ship — a roster grouped your way, a code card with a column the built-in reports do not produce — without rebuilding Scepter.

Start in Report Data Sources: **New** for a working starter script, or **Copy** to begin from one of the shipped ones. The editor runs your script against your real CEOI and shows the rows, so you are never guessing.

The Shape of a Script

```
/**
 * @scepter-datasource
 * id:          MY_ROSTER
 * name:        My Roster
 * entity:      Unit
 * description: Every unit with its parent group.
 */

function build(ceoi, params, props) {

    const rows = [];

    for (const unit of ceoi.getUnits())
        rows.push({
            name:   unit.getName(),
            number: unit.getNumber()
        });

    return rows;
}
```

Header line	Meaning
`id`	Required. What a template saves to point at this data source, so it must be unique and should not change once templates use it. Letters, digits and underscores.
`name`	What appears in the template editor's data source list.
`entity`	The kind of CEOI content the rows describe, which decides which right-click print menus the report appears in: Unit, Group, Location, Frequency, Document, Word, SOI, Sarneg, DRYAD, ChallengeResponse, OneTimePad, Codex, Calendar or CEOI.

`description`	Shown under the list. Say what the rows are.

`build` returns an array of row objects. The keys are the field names the template uses as `$F{...}`. Rows need not all carry the same keys; a field only some rows have simply prints empty on the others.

Honoring the Print Dialog

A data source does not decide what to include on its own. Two independent user choices reach it, and a report that ignores them will not obey the print dialog:

Call	What it carries
`scepter.selectedPeriods()`	The periods chosen in the print dialog. Always a full explicit list, so filter on it rather than testing for empty.
`scepter.selectedDivisions()`	The divisions chosen, same rule.
`scepter.selectedEntities('Location')`	The entities selected by right-clicking a table row selection. Here empty *does* mean all — use `.includes(entity)`, which is true when nothing was selected.
`scepter.selectedIds('Unit')`	The same selection as id strings, for walking rather than testing: `.isEmpty()`, `.contains(id)`, `.ids()`.
`scepter.property(key, fallback)` `scepter.intProperty(key, fallback)`	A property set on the print template or job, such as the condensed code report's `maxLineLength`.

Helpers

The shipped reports share their harder machinery through `scepter`, so a script of your own can reuse it rather than reimplementing it:

Call	What it does
`unitById`, `groupById`, `locationById`, `frequencyById`	Look up an entity by id string; null when the id is unknown.
`groupName(id)`	A group's name, or the standard “unknown group” label.
`entityCodeRows(entity, periods, divisions)`	An entity's codes as a period/division grid, the layout the encode reports use.
`groupMemberships(group, selectedUnits)`	The memberships whose codes a group's reports print.

`actorCodeRows(...)`, `unitCodeRows(...)`	The walk and grid behind the call sign, authentication code and password reports.
`dailyCodeMemberships(...)`, `dailyCodeRows(...)`, `condensedCodeRows(...)`	The same for the two daily code reports.
`selectedGroupsOrAll(...)`	The selected groups, or every group when nothing at all is selected.
`divisionUnits()`, `divisionUnitsWord()`, `maxDivisions()`	The CEOI's division labeling and table width.
`text(key)`	A translated label from Scepter's own string table.
`compareStrings`, `compareIgnoreCase`, `compareNumbers`	Comparators for sorting rows.
`log(message)`	Writes to the application log.

Two Rules About Values

Return plain values, not CEOI objects. A row may hold text, numbers and true/false. Handing back a `Unit` or a `Location` fails with an error naming the field. This is deliberate: Scepter would otherwise serialize that object's whole graph — generated codes included — into the report data, and in a tool that produces cryptographic material that is worth failing loudly over. Read the values you want off the object instead.

A field value cannot be a list. Nested data — a group's members, an SOI's PACE plan — goes in as an array under its own key, and the template reaches it with `((net.sf.jasperreports.json.data.JsonQLDataSource) $P{REPORT_DATA_SOURCE}).subDataSource("members")`. Because it is not a field, a template cannot ask it how many rows it has: if the layout needs that, put a count in the row as well.

What a Script May and May Not Do

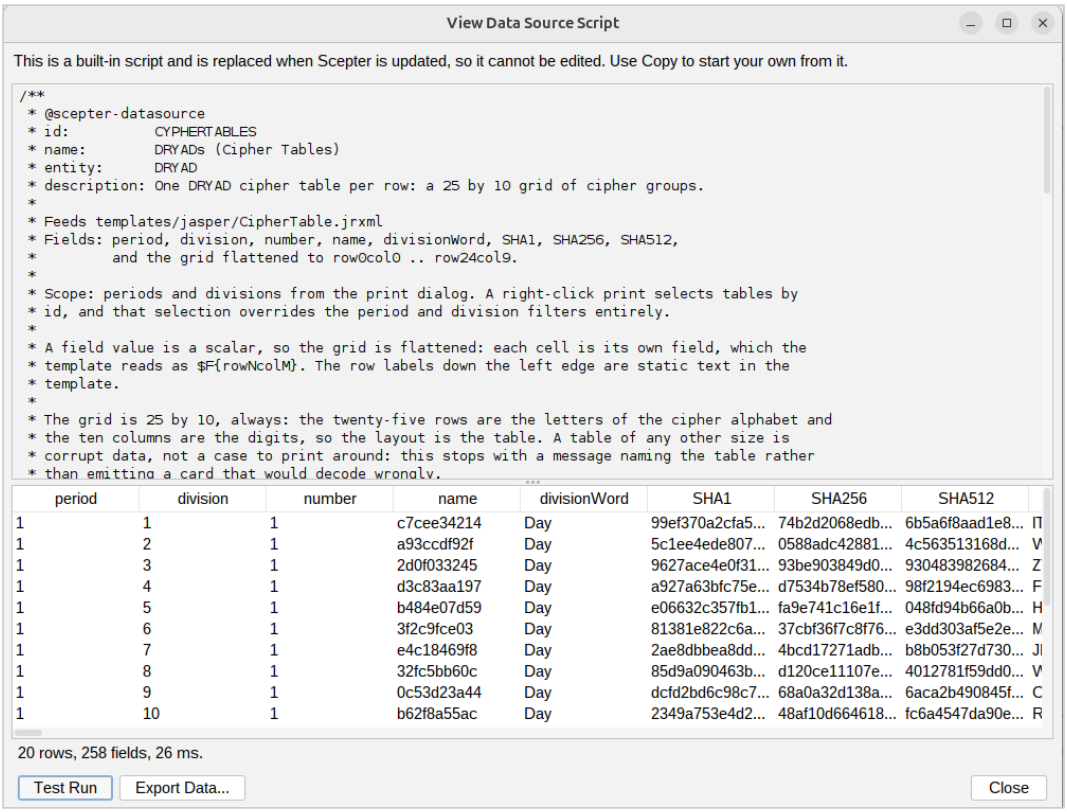
Scripts run in an embedded JavaScript engine with no access to the file system, the network, the Java class path or the ability to start threads or processes. A script can read the open CEOI through the objects it is handed and return values; it cannot reach anything else, so a data source someone sends you cannot read your files or call home.

What it *can* do is put CEOI data on a page. A data source you did not write yourself can print anything the CEOI holds, in any arrangement — so treat one from someone else the way you would treat any document that will be printed with your call signs on it. Read it first; they are short, and **Test Run** shows you exactly what it produces.

A script that runs too long is stopped rather than hanging the print job: 15 seconds in the editor, 30 in a real print run.

Reading a Built-in Script

Every shipped data source can be opened and read, which is the fastest way to see how one is put together. A built-in cannot be edited — it is replaced when Scepter is updated — so use **Copy** to start your own from it. **Test Run** executes the script against the open CEOI and shows the rows it returns, with a count and the time it took.



When Something Goes Wrong

Save refuses a script that does not parse, has no header or id, uses an id already taken, or defines no `build` function, and puts the cursor on the offending line. Errors while running are reported with the script name and line number, in the editor and in the print job, rather than as a blank report.

Related Topics

- Report Data Sources
- Data Source Script Editor
- Data Source Reference
- Print Templates

Exporting Data

File > Export... writes the open CEOI, or part of it, to a file another program can read: a spreadsheet, a JSON file for a script of your own, whatever an export produces. Each kind of file is made by an *export script*, a short JavaScript file like a report data source, except that what it produces is a file rather than report rows.

Exporting

1. Choose **File > Export...**. The list shows every export script, the ones that ship with Scepter and any of your own, with what each one writes.
2. Select one and click **Export...** (or double-click it).
3. If the script asks for them, choose the periods and divisions to export in the Periods and Divisions dialog. Everything is selected to start with. **Cancel** here abandons the export: nothing more is asked and nothing is written.
4. Confirm that you understand the file will hold unencrypted CEOI data. See below.
5. Choose where to save it. The name and file type are filled in from the script; if you type a name without an extension, the script's is added. If the file already exists you are asked before it is replaced.
6. The export runs in the background. When it finishes, Scepter tells you where the file went and how large it is.

An export covers the **whole CEOI** — every period, every division, every entity — unless its script asks which periods and divisions to export. Those that ask show **Asks for Periods** ticked in Export Scripts; of the shipped ones, Unit Codes (CSV) does.

If an export fails

A script that stops with an error is reported the same way a failed print is: the script's name, the line it stopped on, and the message. Nothing is written in that case. The export goes to a temporary file next to the one you chose, and replaces it only once the script has finished, so a failure — or a script stopped for taking longer than 30 seconds, or for writing more than 100 MB — leaves any file that was already there exactly as it was.

Handle exported files carefully

> An exported file contains live CEOI data — whatever the export writes, which may include call signs, authentication codes and passwords — and it is written **unencrypted**. Treat it exactly as you would a printed CEOI: keep it on media you control, and delete it once it has served its purpose.

Shipped Exports

Export	Writes
Units (CSV)	Every unit, one row each, for a spreadsheet.

Units (JSON)	Every unit as a JSON array, for another program.
Unit Codes (CSV)	Every call sign, authentication code and password of every unit, for every group, period and division, one row per code.

To add your own, or change what a shipped one writes, see Export Scripts and Writing Export Scripts.

> **File > Export...** is not the same as **Tools > Export List...**. Export List writes the rows of the table you are looking at (units, locations, words and so on) in the format Import List reads back. File > Export runs an export script over the CEOI and writes whatever that script produces.

Related Topics

- Export Scripts
- Writing Export Scripts
- Main Window

Periods and Divisions

Some export scripts ask which periods and divisions to export before they run — for example Unit Codes (CSV), so you can export one week's codes rather than the whole CEOI. This dialog appears straight after you choose such a script in File > Export, and when you use **Test Run** on one in the editor.

Choosing

- **Periods** lists every period in the CEOI; the second list every division, labeled with this CEOI's word for them (days, shifts and so on).
- Every period and division is **selected when the dialog opens**, so pressing OK straight away exports everything.
- Click to select one, `Ctrl`+click to add or remove one, `Shift`+click to select a range. **All** and **None** under each list select or clear the whole list.
- **OK** is available only while at least one period and one division are selected.

Canceling

Cancel, `Esc` or closing the window abandons the export: you are not asked anything further, the script does not run, and no file is written. In the editor it cancels the Test Run.

Related Topics

- Exporting Data
- Writing Export Scripts
- Export Scripts

Export Scripts

File > Settings > Export Scripts... lists the scripts behind File > Export...: the ones that ship with Scepter and any you have written yourself. Each is a small JavaScript file that reads the open CEOI and writes a file. Because they are files rather than compiled code, you can read any of them, copy one as a starting point, and write your own without rebuilding Scepter.

Table Columns

Column	Description
Name	The name shown in the File > Export... list.
Id	The script's unique identifier. A file in `config/exports` that declares the same id as a built-in script takes the built-in's place.
Extension	The file type the export writes, used to fill in the file name and filter when you save.
Asks for Periods	Ticked when the script asks which periods and divisions to export before it runs. Change it in the editor.
Source	**Built-in** ships with Scepter, in `templates/exports`, and is replaced on update. **Mine** is yours, in `config/exports`, and is never touched by an update.

Actions

- **New** — opens the editor on a working starter script that writes a small CSV file, so a Test Run shows output straight away.
- **Edit** / **View** — opens the selected script. Your own scripts open for editing; built-in ones open read-only, because an update would overwrite any change.
- **Copy** — starts a new script of your own from the selected one. This is the way to base an export on a built-in.
- **Delete** — removes one of your scripts. Built-in scripts cannot be deleted.

Double-clicking a row opens it, the same as Edit or View.

Related Topics

- Export Script Editor
- Writing Export Scripts
- Exporting Data

Export Script Editor

An export script reads the open CEOI and writes a file. This editor lets you write one, run it against your real data, and see what it would write before you ever save a file.

Ask for periods and divisions

Tick **Ask for periods and divisions when this export is run** to have File > Export show the Periods and Divisions dialog before running the script. The check box and the script's `promptForPeriodDivision:` header line are the same setting: ticking it writes `true` (adding the line if the header has none), clearing it writes `false`, and editing the line by hand moves the check box. It is grayed out for a built-in script, which cannot be edited.

Test Run

Test Run runs the script against the open CEOI exactly as File > Export would — same sandbox, same limits — but keeps the output in memory instead of writing it to disk. If the script asks for periods and divisions, Test Run asks too, so you can try it on a narrowed selection; canceling there cancels the Test Run. The lower half of the window shows the start of what it wrote:

- **Text** output (anything that is valid UTF-8) is shown as text, up to the first 64 KB.
- **Binary** output is shown as a hex dump of its first 4 KB: the offset, the bytes in hexadecimal, and the printable characters among them.

The status line gives the size of the whole file the export would write and how long it took. A Test Run is stopped after 15 seconds; a real export is allowed 30.

Saving

Save refuses a script that does not parse, has no `@scepter-export` header or id, uses an id already taken, gives an extension that is not plain letters and digits, or defines no `exportData` function, and puts the cursor on the offending line where there is one. A saved script appears under File > Export straight away.

Built-in scripts open read-only, because they are replaced when Scepter is updated. Use **Copy** in the Export Scripts list to start your own from one.

Related Topics

- Writing Export Scripts
- Export Scripts
- Exporting Data

Writing Export Scripts

An export script is a JavaScript file that reads the open CEOI and writes a file. Writing one lets you get CEOI data into another program — a spreadsheet laid out your way, a JSON file for a script of your own, a binary format some device expects — without rebuilding Scepter.

Start in Export Scripts: **New** for a working starter script, or **Copy** to begin from one of the shipped ones. The editor's **Test Run** shows what your script would write, so you are never guessing.

The Shape of a Script

```
/**
 * @scepter-export
 * id:          UNITS_CSV
 * name:        Units (CSV)
 * extension:   csv
 * description: Every unit with its number.
 */

function exportData(ceoi, params, props, out) {
    out.writeLine("name,number");
    for (const unit of ceoi.getUnits())
        out.writeLine(unit.getName() + "," + unit.getNumber());
}
```

Header line	Meaning
`@scepter-export`	Marks the file as an export script. A report data source carries `@scepter-datasource` instead, and neither kind appears in the other's list.
`id`	Required and unique. Letters, digits and underscores. The editor will not save a script under an id already in use, but a file you put in `config/exports` yourself with a built-in's id takes the built-in's place under File > Export.
`name`	What the File > Export list shows.
`extension`	The file type, without the dot: used for the Save As filter and the suggested file name. Letters and digits only.
`promptForPeriodDivision`	`true` to have File > Export ask which periods and divisions to export before running the script; `false` or no line at all to cover them all. Only `true` and `false` are accepted. The editor's **Ask for periods and divisions** check box sets this line for you.
`description`	

	Shown under the name in the File > Export list. Say what the file holds, in one line: only the line the `description:` is on is read, so put any longer explanation in the comment below the header.

`exportData` is the one function a script must define. It returns nothing: whatever it writes to `out` is the file. Your own scripts live in `config/exports`; the shipped ones are in `templates/exports`.

Writing the File: `out`

Call	What it writes
`out.write(text)`	Text, encoded as UTF-8. Anything that is not a string is written as JavaScript's `String()` of it.
`out.writeLine(text)`	Text followed by a newline. Lines always end with a single `\n`, whatever the computer, so the same CEOI gives the same file everywhere. `out.writeLine()` on its own writes just the newline.
`out.writeBytes(bytes)`	Raw bytes, for a binary format: a `Uint8Array`, an `ArrayBuffer`, or an ordinary array of whole numbers from 0 to 255. A value outside that range stops the export with an error naming its position, rather than being quietly turned into a different byte.

Text and bytes can be mixed in one file, in the order they are written. `out` has no other methods: a script cannot find out where its file is going, and cannot finish or discard it — Scepter does that once `exportData` returns.

The Rest of What a Script Sees

Everything a data source has is here too: the `ceoi` object, and the `scepter` helpers such as `groupById`, `unitById`, `compareStrings` and `log`. There is never a right-click selection.

Periods and Divisions

`scepter.selectedPeriods()` and `scepter.selectedDivisions()` return arrays of period and division numbers. For a script whose header says `promptForPeriodDivision: true` they hold what the user chose in the Periods and Divisions dialog; for any other script they hold every period and division in the CEOI. The same choice is in `props.periods` and `props.divisions` as comma-separated text. A script that asks should filter on them:

```
const periods = scepter.selectedPeriods();
const divisions = scepter.selectedDivisions();

for (const code of membership.getCallsigns())
    if (periods.includes(code.getPeriod()) && divisions.includes(code.getDivision()))
        out.writeLine(code.getCode());
```

The user cannot pick nothing: the dialog will not let them press OK with an empty list, and canceling it abandons the export before your script runs.

CSV is the commonest format and the easiest to get subtly wrong. A field that holds a comma, a double quote or a line break has to be put in double quotes, with any quotes inside doubled, or a spreadsheet will split the row. The shipped Units (CSV) script has a five-line `csv(value)` function that does this; copy it.

What a Script May and May Not Do

An export script runs in exactly the same sandbox as a data source. It has no access to the file system, the network, the Java class path, or the ability to start threads or processes, and the Java objects it is handed expose their data but not the machinery behind them (reflection, class loaders, files, streams). The *only* thing an export script has that a data source does not is `out`, and that writes only to the one file the user chose in Save As.

What it *can* do is put any CEOI data it likes into that file. An export script you did not write yourself could write call signs, codes and passwords into a file you then hand to someone else — so read one from elsewhere before you use it. They are short, and Test Run shows you exactly what they produce.

Limit	What happens
Time	Stopped after 30 seconds (15 in a Test Run), so a script stuck in a loop cannot hang Scepter.
Size	Stopped at 100 MB, so a runaway loop cannot fill the disk or USB stick it is writing to.
Failure	An error, the time limit or the size limit all leave no file behind, and leave any file already at that name untouched.

Reading a Built-in Script

Every shipped export can be opened and read, which is the fastest way to see how one is put together. A built-in cannot be edited — it is replaced when Scepter is updated — so use **Copy** to start your own from it.

When Something Goes Wrong

Errors are reported with the script's name and line number, both in the editor's Test Run and when File > Export runs it. `scepter.log(message)` writes to the application log if you need to see what a script is doing part way through.

Related Topics

- Export Scripts
- Export Script Editor
- Exporting Data
- Writing Custom Data Sources

Export: Units (CSV)

Writes every unit in the CEOI to a CSV file, one row per unit, for opening in a spreadsheet.

Export Details

Field	Value
Script File	`templates/exports/units-csv.js`
Id	UNITS_CSV
File Type	`.csv`, UTF-8, lines ending in `\n`

Columns

Column	Contents
name	The unit's name.
number	The unit's number.
Field A, B and C	The unit's three custom fields, headed with the names this CEOI gives them on the Units options page (`field_a`, `field_b` and `field_c` where a field has no name).
groups	The groups the unit belongs to, separated by `;`.
notes	The unit's notes.

Rows are sorted by unit name, after a header row. A value holding a comma, a double quote or a line break is quoted as spreadsheets expect, so every unit reads back as exactly one row.

> Copy this script in Export Scripts to add columns — a unit's call signs, say — or to change the separator for a program that expects semicolons.

Related Topics

- Exporting Data
- Units (JSON)
- Writing Export Scripts

Export: Units (JSON)

Writes every unit in the CEOI to a JSON file, for reading into another program or script.

Export Details

Field	Value
Script File	`templates/exports/units-json.js`
Id	UNITS_JSON
File Type	`.json`, UTF-8, indented two spaces

Format

The file is an array with one object per unit, sorted by unit name:

```
[
  {
    "name": "Dale Whitcomb",
    "number": 3,
    "fields": {
      "Role": "Operations Chief",
      "Field B": null,
      "Field C": null
    },
    "groups": [ "All Units Group", "Leadership Group" ],
    "notes": null
  }
]
```

Key	Contents
name	The unit's name.
number	The unit's number.
fields	The unit's three custom fields, keyed by the names this CEOI gives them (`field_a`, `field_b` and `field_c` where a field has no name). An empty field is `null`.
groups	The names of the groups the unit belongs to.
notes	The unit's notes, or `null`.

Related Topics

- Exporting Data

- Units (CSV)
- Writing Export Scripts

Export: Unit Codes (CSV)

Writes every code every unit holds — call signs, authentication codes and passwords, in every group the unit belongs to — to a CSV file, one row per code. Before it runs it asks which periods and divisions to include; everything is selected to start with.

Export Details

Field	Value
Script File	`templates/exports/unit-codes-csv.js`
Id	UNIT_CODES_CSV
File Type	`.csv`, UTF-8, lines ending in `\n`
Asks for Periods	Yes

Columns

Column	Contents
group	The group the code belongs to. A unit's codes are generated per group membership, so a unit in two groups has two sets. Blank if the group no longer exists.
unit	The unit's name.
unit_number	The unit's number.
type	`callsign`, `authcode` or `password`.
period	The period the code is for.
division	The division within the period.
number	Which code this is when a unit has several of one type in the same period and division.
code	The code itself.

Rows are sorted by group, unit, type (call signs, then authentication codes, then passwords), period, division and number. One row per code makes the file easy to filter, or to pivot into a period-by-division grid, in a spreadsheet. Only codes in the periods and divisions you chose are written.

> This file holds every call sign, authentication code and password in the CEOI, unencrypted. It is as sensitive as the complete printed code set: keep it on media you control and delete it when it has served its purpose.

Related Topics

- Exporting Data
- Units (CSV)
- Writing Export Scripts

Reports Overview

Scepter produces printed CEOI output through a templating system. Each **Print Template** is a file that defines the layout and content of one page type — call signs, OTP pads, cipher tables, and so on. Templates are selected and ordered in a Print Configuration, which is then run as a Print Job.

Template Types

Two template formats are supported:

Type	File Extension	Description
Jasper Reports	`.jrxml`	The primary format. Report layouts are designed visually in Jasper Studio (a free Eclipse-based IDE). The `.jrxml` file is compiled by Scepter at print time. Supports multi-page output, sub-reports, tables, banding, and rich formatting.
FreeMarker	`.ftlh`	A plain-text template language. FreeMarker templates produce text or HTML output and are easy to hand-edit in any text editor. Well-suited for simple cover sheets or structured text output.

Where Templates Live

The built-in templates are shipped in the `templates/` folder inside the Scepter installation directory:

Folder	Contents
`templates/jasper/`	All `.jrxml` Jasper Reports templates
`templates/freemarker/`	All `.ftlh` FreeMarker templates

Scepter searches for template files by name starting from the folder shown in each template's **Template File** field. You can place additional template files anywhere on disk and point to them by full path.

Customizing Templates

> **Always copy a built-in template before modifying it.** Updates to Scepter may overwrite files in the `templates/` folder, discarding any changes you made there directly.

1. Copy the `.jrxml` or `.ftlh` file to a folder outside the Scepter installation (e.g., a `my-templates/` folder in your persistent storage on Tails).
2. Edit the copy in Jasper Studio or a text editor.
3. In Print Templates, create a new template entry and set its **Template File** to the full path of your copy.
4. Add the new template to a Print Configuration in place of (or alongside) the original.

Data Sources

Each template is bound to a **Data Source** type that controls what CEOI data is passed to it. The data source is set when defining the template in Print Templates and cannot be changed without also changing the report layout. The available data sources are listed on each individual template page.

Template Properties

Some templates accept optional **Properties** — key/value pairs entered in the template definition — that control runtime behavior (e.g., number of groups per line, page orientation, or direction). Each template page below documents the properties it accepts.

Available Templates

Template	Type	Produces
Authentication Codes	Jasper	Unit authentication codes by period
Basic Text Cover Sheet	FreeMarker	Plain-text cover sheet with title and distribution
Call Signs	Jasper	Unit callsigns, landscape layout
Challenge Response Words	Jasper	Challenge/response/duress word pairs
Cover Sheet (pdf)	Jasper	Formatted PDF cover page for the CEOI package
DRYADs	Jasper	DRYADs
Daily Codes (Multi Divisions Per Page)	Jasper	Combined daily code card, multiple divisions per page
Daily Codes (Single Division Per Page)	Jasper	Daily code card, one division per page
Daily Codes Condensed	Jasper	Compact daily codes, one division per page
Document Codes – Decrypt	Jasper	Document code-to-name lookup table
Document Codes – Encrypt	Jasper	Document name-to-code lookup table
Frequency Codes – Decrypt	Jasper	Frequency code-to-name lookup table
Frequency Codes – Encrypt	Jasper	Frequency name-to-code lookup table
Group Roster	Jasper	Group membership roster
Location Codes – Decrypt	Jasper	Location code-to-coordinate lookup table
Location Codes – Encrypt	Jasper	Location coordinate-to-code lookup table
Month Calendar	Jasper	Landscape monthly calendar
OTP DIANA Conversion Table	Jasper	DIANA letter-to-digit codex table

OTP Diana Plus Conversion Table	Jasper	DIANA+ reciprocal codex table
OTP Numeric Conversion Table	Jasper	Numeric OTP codex table
One Time Pad	Jasper	Standard OTP pad sheet
One Time Pad IN	Jasper	Directional OTP pad for incoming messages
One Time Pad IN Simplified	Jasper	Simplified incoming OTP pad, 5 groups per line
One Time Pad OUT	Jasper	Directional OTP pad for outgoing messages
One Time Pad OUT Simplified	Jasper	Simplified outgoing OTP pad, 5 groups per line
Passwords	Jasper	Unit passwords, landscape layout
SARNEGs	Jasper	SARNEG cipher grid tables
SOI	Jasper	Signals Operating Instruction sheet
Unit Detail	Jasper	Full unit information sheet
Wordlist Decode	Jasper	Word code-to-plaintext lookup table
Wordlist Encode	Jasper	Plaintext-to-word code lookup table

Related Topics

- Print Templates
- Print Configurations
- Print Jobs

Report: Basic Text Cover Sheet

```
OPERATION EXAMPLE — CEOI
Official Use Only


---------------------------------------------
```

Produces a simple plain-text cover sheet containing the CEOI title and distribution line. Because it uses the FreeMarker template engine, it can be edited in any text editor without Jasper Studio.

Template Details

Field	Value
Template File	`coversheet.ftlh`
Template Type	FreeMarker
Data Source	CEOI
Properties	*None*

Output

A short text block with the CEOI title, distribution, and a separator line. This is the simplest cover sheet available and is suitable for plain-text or low-classification packaging.

Customizing

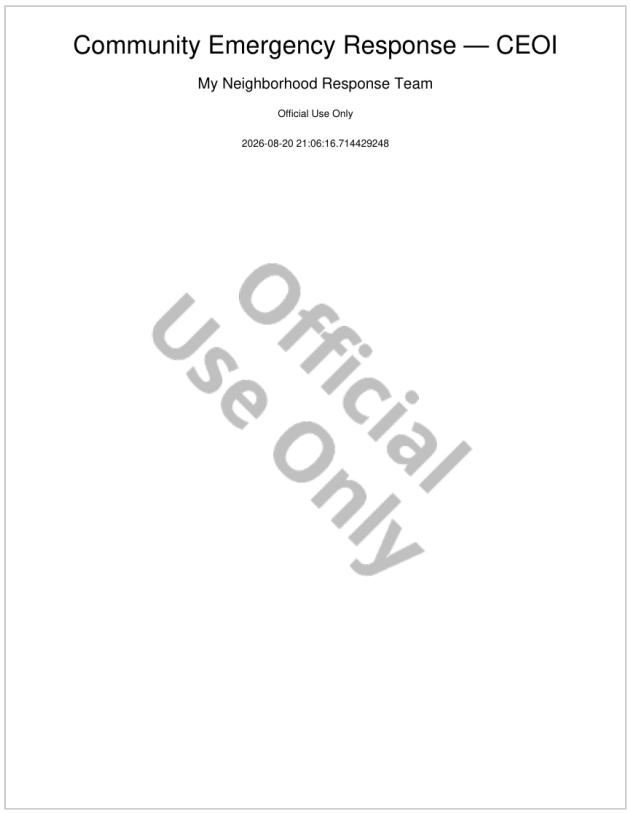
The template uses FreeMarker variable references such as `${'$'}{CEOI.title}` and `${'$'}{CEOI.distribution}`. Copy `templates/freemarker/coversheet.ftlh` to your own folder and add any additional text, headers, or FreeMarker expressions you need.

> FreeMarker templates are plain text files — no special IDE is required. A standard code editor is sufficient.

Related Topics

- Reports Overview
- Cover Sheet (pdf)
- Print Templates

Report: Cover Sheet (pdf)



Produces a formatted PDF cover page for the CEOI package. Includes the CEOI title, period dates, classification, and distribution information in a styled layout suitable for placing at the front of a printed CEOI.

Template Details

Field	Value
Template File	`CoverPage.jrxml`
Template Type	Jasper Reports
Data Source	CEOI
Properties	*None*

Output

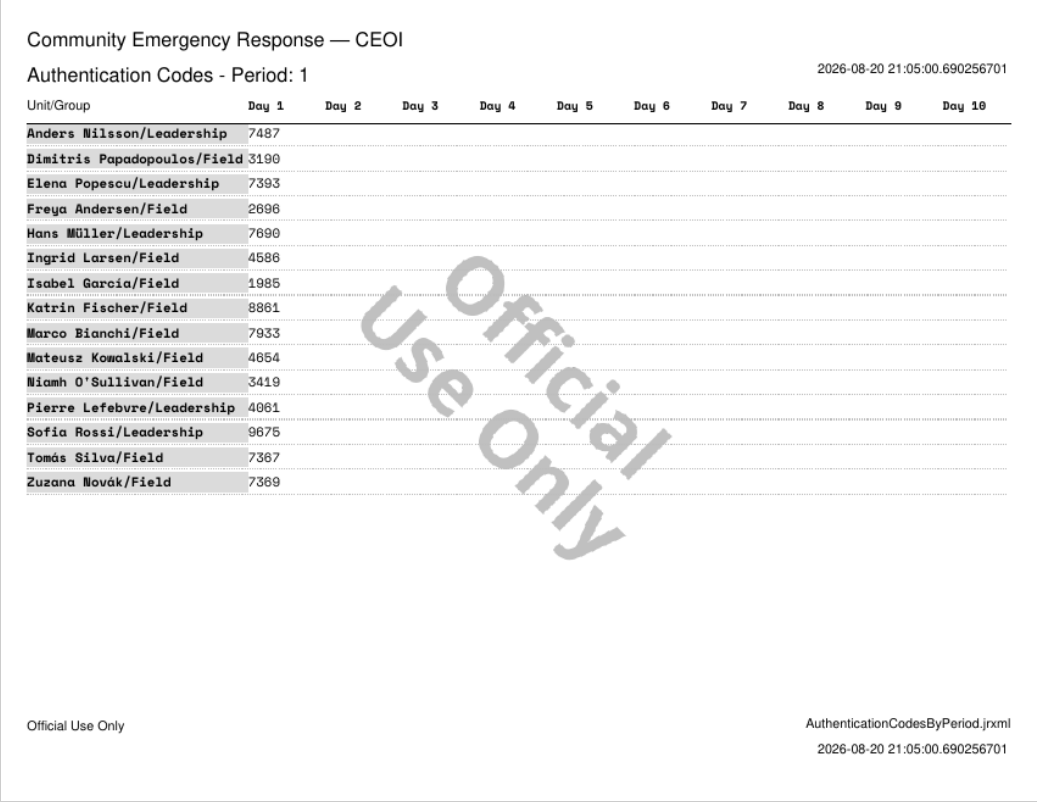
A single styled cover page. Fields are populated from the CEOI document properties (title, distribution, dates). Designed to be the first page of a Print Configuration.

> Copy `templates/jasper/CoverPage.jrxml` and open it in Jasper Studio to add unit insignia, change fonts, or restructure the layout for your unit's standard format. For a plain-text alternative, see Basic Text Cover Sheet.

Related Topics

- Reports Overview
- Basic Text Cover Sheet
- Cover Settings
- Print Templates

Report: Authentication Codes


Community Emergency Response — CEOI

Authentication Codes - Period: 1

2026-08-20 21:05:00.690256701

Unit/Group	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Anders Nilsson/Leadership	7487									
Dimitris Papadopoulos/Field	3190									
Elena Popescu/Leadership	7393									
Freya Andersen/Field	2696									
Hans Müller/Leadership	7690									
Ingrid Larsen/Field	4586									
Isabel García/Field	1985									
Katrin Fischer/Field	8861									
Marco Bianchi/Field	7933									
Mateusz Kowalski/Field	4654									
Niamh O'Sullivan/Field	3419									
Pierre Lefebvre/Leadership	4061									
Sofia Rossi/Leadership	9675									
Tomás Silva/Field	7367									
Zuzana Novák/Field	7369									

Official Use Only

Official Use Only

AuthenticationCodesByPeriod.jrxml

2026-08-20 21:05:00.690256701

Prints a table of unit authentication codes organized by period and division. Each row shows the unit name, period, division, and the generated authentication code assigned to that unit for that time window.

Template Details

Field	Value
Template File	`AuthenticationCodesByPeriod.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_AUTHENTICATION_CODES_BY_PERIOD
Properties	*None*

Output

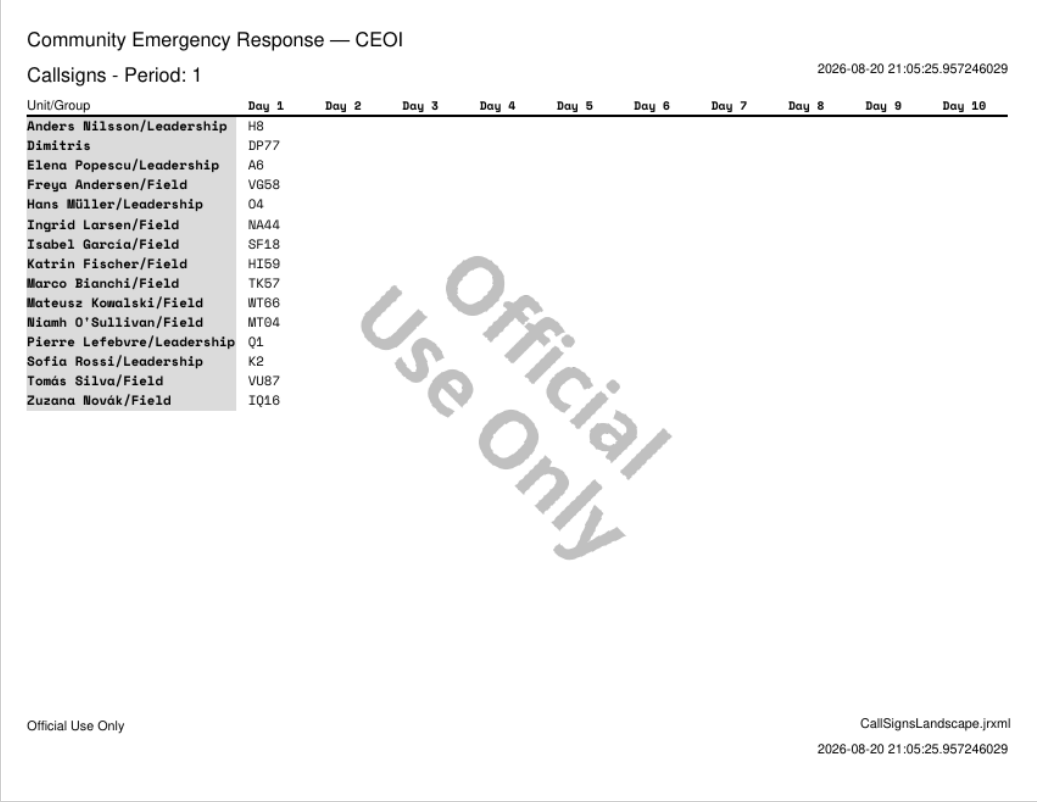
One report covering all units across all periods and divisions in the CEOI. Rows are grouped by period then division, making it easy to distribute one page per time window.

> To customize fonts, column widths, or add a classification header, copy `templates/jasper/AuthenticationCodesByPeriod.jrxml` to your own templates folder and open it in Jasper Studio. See Reports Overview for the recommended workflow.

Related Topics

- Reports Overview
- Unit Auth Codes
- Print Templates

Report: Call Signs


Community Emergency Response — CEOI

Callsigns - Period: 1

2026-08-20 21:05:25.957246029

Unit/Group	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Anders Nilsson/Leadership	H8									
Dimitris	DP77									
Elena Popescu/Leadership	A6									
Freya Andersen/Field	VG58									
Hans Müller/Leadership	O4									
Ingrid Larsen/Field	NA44									
Isabel García/Field	SF18									
Katrin Fischer/Field	HI59									
Marco Bianchi/Field	TK57									
Mateusz Kowalski/Field	WT66									
Niamh O'Sullivan/Field	MT04									
Pierre Lefebvre/Leadership	Q1									
Sofia Rossi/Leadership	K2									
Tomás Silva/Field	VU87									
Zuzana Novák/Field	IQ16									

Official Use Only

Official Use Only

CallSignsLandscape.jrxml

2026-08-20 21:05:25.957246029

Prints a landscape-format table of unit callsigns, organized by period and division. Each row shows the unit name and its assigned callsign for that time window.

Template Details

Field	Value
Template File	`CallSignsLandscape.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_CALLSIGNS
Properties	`landscape=true`

Output

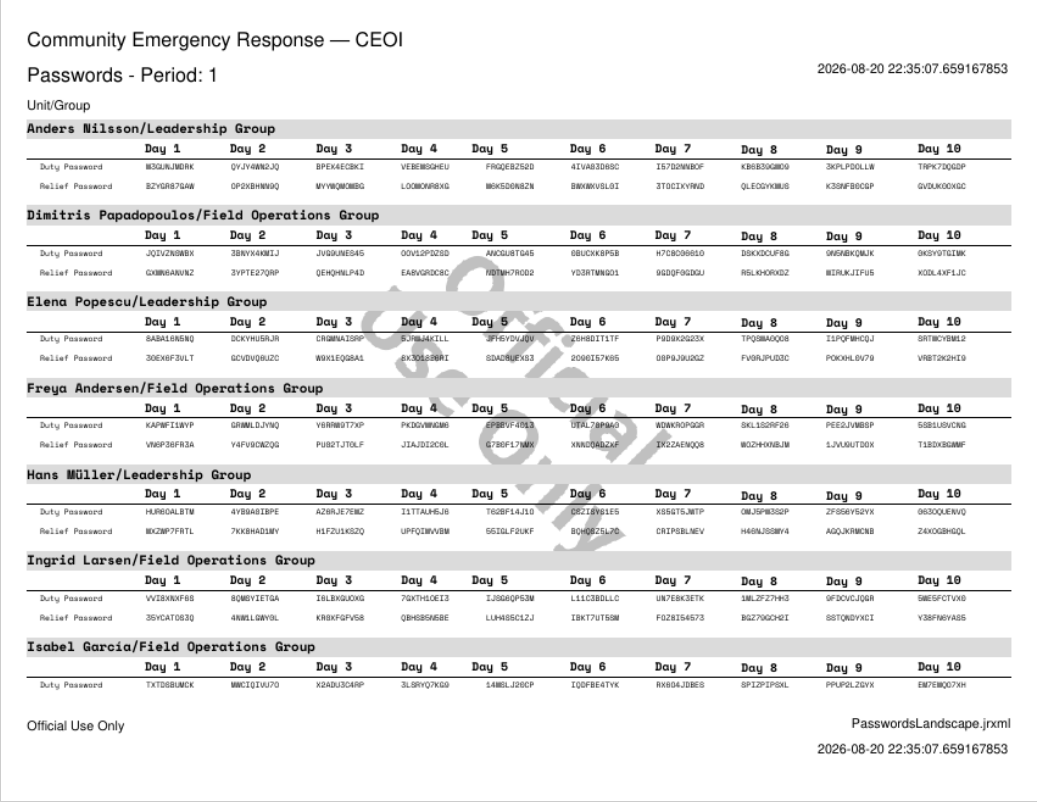
Landscape-oriented pages listing all unit callsigns across all periods and divisions. Columns are sized to fit the landscape format for easy field reference.

> Copy `templates/jasper/CallSignsLandscape.jrxml` to customize column widths, add group callsigns, or change the page orientation.

Related Topics

- Reports Overview
- Unit Callsigns
- Print Templates

Report: Passwords



Community Emergency Response — CEOI

Passwords - Period: 1

2026-08-20 22:35:07.659167853

Unit/Group

Anders Nilsson/Leadership Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	W3GUNJWDRK	QVJY4WN2JQ	BPEX4ECBKI	VEBEWBGHEU	FRQQEBZ52D	4IVA83D8SC	I57D2NNBOF	KB6B3QGWO9	3KPLPDOLLW	TRPK7DQGDP
Relief Password	BZYGR87GAW	OP2XBHNW9Q	MYYWQWOWBG	LOOMOWRBXG	WGK5D6N8ZN	BWKWKVSL0I	3T0CIKYRWD	QLECGYKMUS	K3BNFB0CGP	GVDUKOOXGC

Dimitris Papadopoulos/Field Operations Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	JQIVZN0WBX	3BNYX4KWIJ	JV0QUNES4S	OOV12PDZSD	ANCQU8TG45	0BUCXK8P5B	H7CBC0861O	DSKXDCUF8G	0W5NBKQWJK	0K5Y0TGIWK
Relief Password	GXWW6ANVNZ	3YPTE27QRP	QEHQHNLP4D	EA8VGRDC8C	NDTWH7ROD2	YD3RTWNQO1	0GDQF0GDGU	R6LKHORXDZ	WIRUKJIFU5	KODL4XF1JC

Elena Popescu/Leadership Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	8ABA186N8Q	DCKYHU5RJR	CRQWNAISRP	5JRRJ4KILL	JPH6YDVJQV	Z6H8DIT1TF	P0D0K2G23X	TPQ5WAQQO8	I1PQFWHCQJ	SRTWCYBWL2
Relief Password	36EX8F3VLT	GCVDVQ8U2C	W9K1EQG8A1	6K3OU88BRI	SDADBUEX83	2O96I67K6S	O8P9JGU2GZ	FV0RJPUD3C	POKXHL0V79	VRBT2K2HI9

Freya Andersen/Field Operations Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	KAPWFI1WVP	GRWWLDJYWQ	V6RRW9T7XP	PKDQVWWGW6	EP3BUF4613	UTAL78R9AS	WDWKROPQQR	SKL1S2RF26	PEE2JVWB8P	5GB1U9VCNG
Relief Password	VN6P38FR3A	Y4FV9CWZQG	PU82TJTOLF	JIAJDI2C6L	G7B0F17NWX	XNNDOADZXF	IX22AENQQ8	WO2HHXNBJM	1JVU0UTD0X	T1B2XBGWWF

Hans Müller/Leadership Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	HUR6OALBTW	4VB9A0IBPE	AZ6RJE7EWZ	I1TTAUH6J8	T62BF14J1O	C6ZI6YG1E5	X85GT5JWTP	OWJSPW3S2P	ZFS66Y52YX	063OQUENVQ
Relief Password	WXZWP7FRTL	7KKBHAD1MY	H1FZU1KSZQ	UPFQIMVVBW	55IGLF2UKF	BQHQ6Z5L7O	CRIPSBLNEV	H40NJ8S8MY4	AGQJKRWCNB	Z4XOGBHGQL

Ingrid Larsen/Field Operations Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	VVI8XNXF6S	8QW8YIETQA	I6LBKQUOXG	7GXTH1OEI3	IJ8G6QPS3W	L11C3BDLLC	UN7E8K3ETK	1WLZFZ7HK3	9FDCVCJQ6R	6WE5FCTVX0
Relief Password	35YCATD83Q	4NW1LGWY0L	KR8KFGFVS8	QBHS85W5BE	LUH48SC1ZJ	IBKT7UT58W	FOZ8IS4573	BGZ79GCH2I	SSTQWDYXCI	Y38FN6YAB5

Isabel García/Field Operations Group

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Duty Password	TXTD8BUWCK	WWCIQIVU7O	X2ADU3CARP	3L5RYQ7KG9	14W0LJJ0CP	IQDFBE4TYK	RX8G4JDBES	SPI2PIP9XL	PPUP2LZGYX	EW7EWQO7XH

Official Use Only

PasswordsLandscape.jrxml

2026-08-20 22:35:07.659167853

Prints a landscape-format table of unit passwords, organized by period and division. Each row shows the unit name and its assigned password for that time window.

Template Details

Field	Value
Template File	`PasswordsLandscape.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_PASSWORDS
Properties	*None*

Output

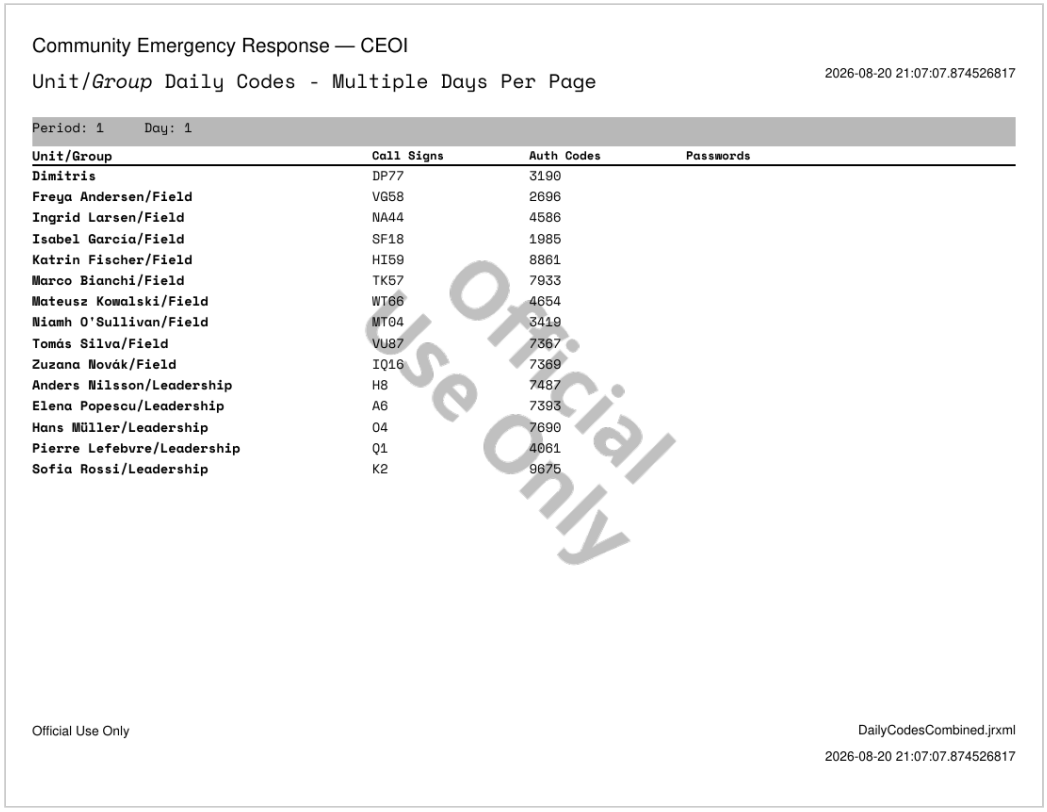
Landscape-oriented pages listing all unit passwords across all periods and divisions. Layout matches the Call Signs report for a consistent set of COMSEC cards.

Related Topics

- Reports Overview
- Unit Passwords
- Call Signs

- Print Templates

Report: Daily Codes (Multi Divisions Per Page)



Community Emergency Response — CEOI

Unit/*Group* Daily Codes - Multiple Days Per Page

2026-08-20 21:07:07.874526817

Period: 1 Day: 1

Unit/Group	Call Signs	Auth Codes	Passwords
Dimitris	DP77	3190	
Freya Andersen/Field	VG58	2696	
Ingrid Larsen/Field	NA44	4586	
Isabel García/Field	SF18	1985	
Katrin Fischer/Field	HI59	8861	
Marco Bianchi/Field	TK57	7933	
Mateusz Kowalski/Field	WT66	4654	
Niamh O'Sullivan/Field	MT04	3419	
Tomás Silva/Field	VU87	7367	
Zuzana Novák/Field	IQ16	7369	
Anders Nilsson/Leadership	H8	7487	
Elena Popescu/Leadership	A6	7393	
Hans Müller/Leadership	O4	7690	
Pierre Lefebvre/Leadership	Q1	4061	
Sofia Rossi/Leadership	K2	9675	

Official Use Only

Official Use Only

DailyCodesCombined.jrxml

2026-08-20 21:07:07.874526817

Prints daily code cards with multiple divisions combined on each page. This layout is more compact than the single-division version and is useful when a period covers many short divisions (e.g., four six-hour shifts per day).

Template Details

Field	Value
Template File	`DailyCodesCombined.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_DAILY_CODES
Properties	*None*

Output

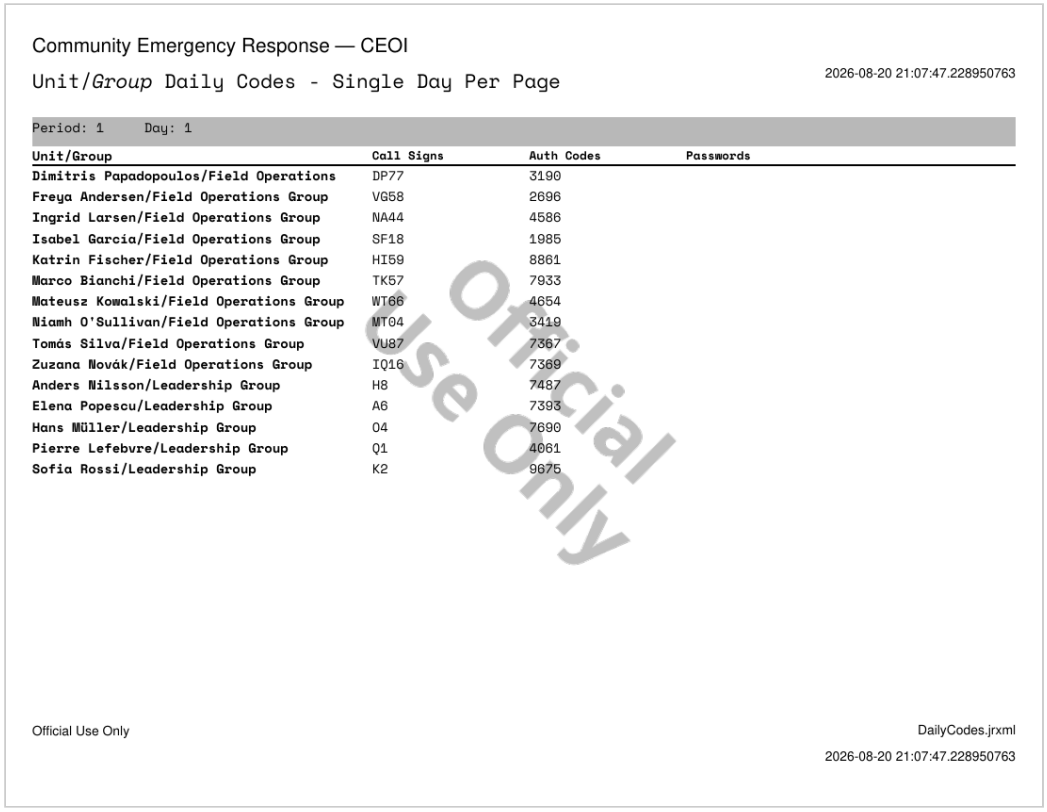
Daily code cards for all units, with multiple divisions stacked on each page. Callsigns, authentication codes, and passwords are shown side by side for each unit and division.

> For one division per page, use Daily Codes (Single Division Per Page) instead. For a more compact layout, see Daily Codes Condensed.

Related Topics

- Reports Overview
- Daily Codes (Single Division Per Page)
- Daily Codes Condensed
- Print Templates

Report: Daily Codes (Single Division Per Page)



Community Emergency Response — CEOI

Unit/*Group* Daily Codes - Single Day Per Page

2026-08-20 21:07:47.228950763

Period: 1 Day: 1

Unit/Group	Call Signs	Auth Codes	Passwords
Dimitris Papadopoulos/Field Operations	DP77	3190	
Freya Andersen/Field Operations Group	VG58	2696	
Ingrid Larsen/Field Operations Group	NA44	4586	
Isabel García/Field Operations Group	SF18	1985	
Katrin Fischer/Field Operations Group	HI59	8861	
Marco Bianchi/Field Operations Group	TK57	7933	
Mateusz Kowalski/Field Operations Group	WT66	4654	
Niamh O'Sullivan/Field Operations Group	MT04	3419	
Tomás Silva/Field Operations Group	VU87	7367	
Zuzana Novák/Field Operations Group	IQ16	7369	
Anders Nilsson/Leadership Group	H8	7487	
Elena Popescu/Leadership Group	A6	7393	
Hans Müller/Leadership Group	O4	7690	
Pierre Lefebvre/Leadership Group	Q1	4061	
Sofia Rossi/Leadership Group	K2	9675	

Official Use Only

Official Use Only

DailyCodes.jrxml

2026-08-20 21:07:47.228950763

Prints daily code cards with one division per page. Each page covers all units for a single period and division, showing callsigns, authentication codes, and passwords in a clear layout suitable for distribution as individual sheets.

Template Details

Field	Value
Template File	`DailyCodes.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_DAILY_CODES
Properties	*None*

Output

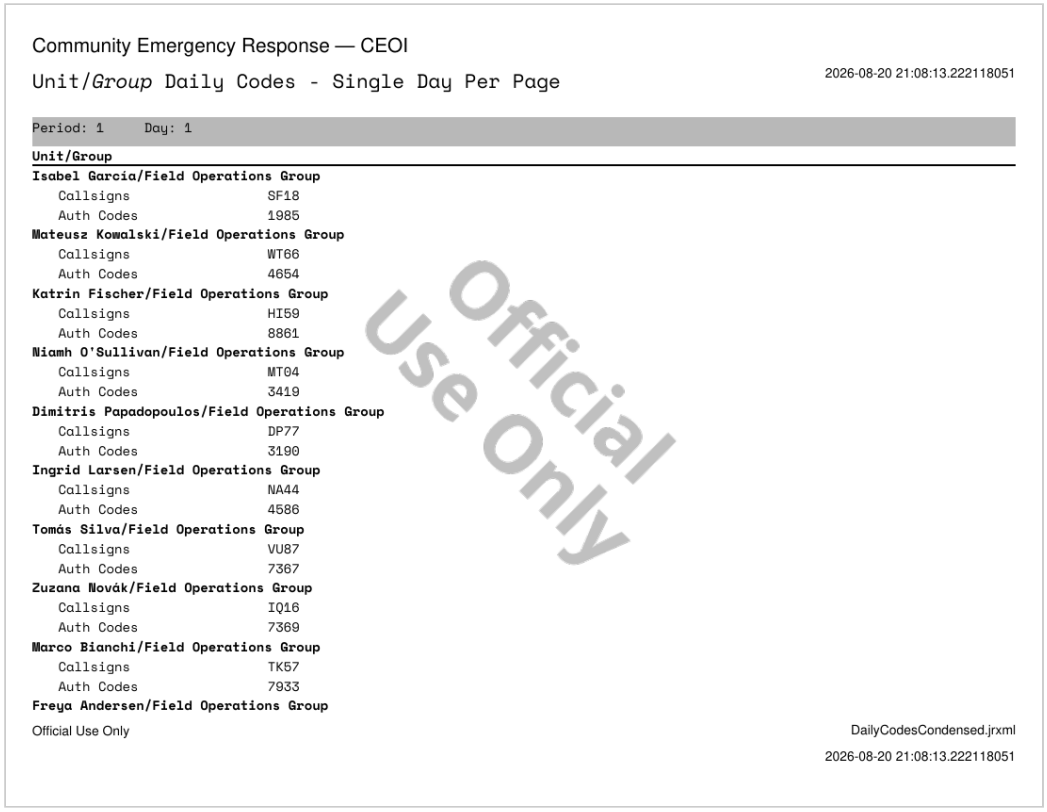
One page per division, listing all units and their codes for that division. Well suited for operations where each time period is distributed separately and the page itself becomes the daily COMSEC card.

> For a more compact layout that fits multiple divisions per page, use Daily Codes (Multi Divisions Per Page).

Related Topics

- Reports Overview
- Daily Codes (Multi Divisions Per Page)
- Daily Codes Condensed
- Print Templates

Report: Daily Codes Condensed


Community Emergency Response — CEOI

Unit/*Group* Daily Codes - Single Day Per Page

2026-08-20 21:08:13.222118051

Period: 1 Day: 1

Unit/Group

Isabel Garcia/Field Operations Group
Callsigns SF18
Auth Codes 1985
Mateusz Kowalski/Field Operations Group
Callsigns WT66
Auth Codes 4654
Katrin Fischer/Field Operations Group
Callsigns HI59
Auth Codes 8861
Niamh O'Sullivan/Field Operations Group
Callsigns MT04
Auth Codes 3419
Dimitris Papadopoulos/Field Operations Group
Callsigns DP77
Auth Codes 3190
Ingrid Larsen/Field Operations Group
Callsigns NA44
Auth Codes 4586
Tomás Silva/Field Operations Group
Callsigns VU87
Auth Codes 7367
Zuzana Novák/Field Operations Group
Callsigns IQ16
Auth Codes 7369
Marco Bianchi/Field Operations Group
Callsigns TK57
Auth Codes 7933
Freya Andersen/Field Operations Group

Official Use Only

Official Use Only

DailyCodesCondensed.jrxml

2026-08-20 21:08:13.222118051

Prints a condensed daily code card with one division per page. Text is formatted more tightly than the standard daily codes layout, fitting more units on a page by wrapping longer code lines to a configurable maximum line length.

Template Details

Field	Value
Template File	`DailyCodesCondensed.jrxml`
Template Type	Jasper Reports
Data Source	UNIT_CODES_CONDENSED
Properties	`maxLineLength=92`

Properties

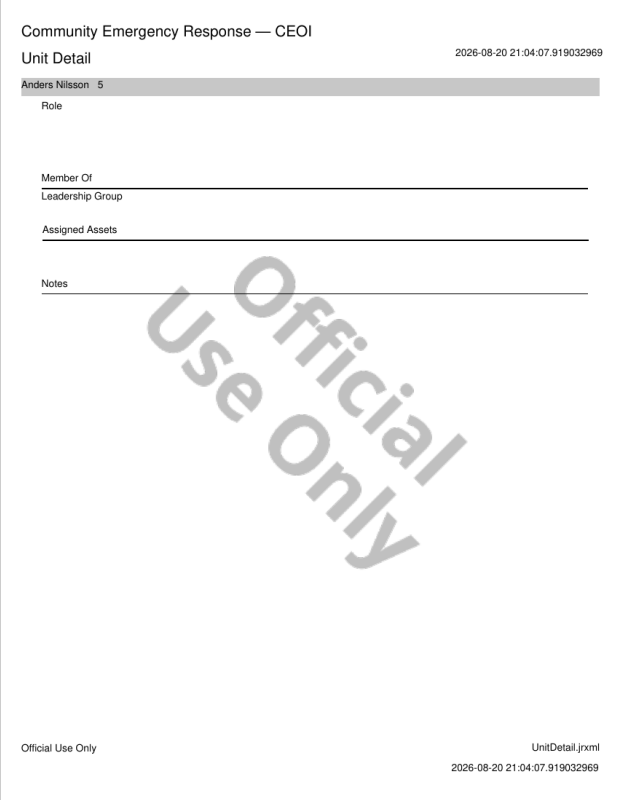
Property	Description
`maxLineLength`	Maximum number of characters per line before the output wraps. Default is `92`. Reduce this value for narrower paper or larger fonts; increase it to fit more data per line.

> Copy `templates/jasper/DailyCodesCondensed.jrxml` to adjust font size or margins if you are printing to a non-standard paper size.

Related Topics

- Reports Overview
- Daily Codes (Single Division Per Page)
- Daily Codes (Multi Divisions Per Page)
- Print Templates

Report: Unit Detail



Prints a full detail sheet for each unit in the CEOI. Each sheet includes the unit name, settings, group memberships, all generated codes (callsigns, authentication codes, passwords), and assigned crypto assets.

Template Details

Field	Value
Template File	`UnitDetail.jrxml`
Template Type	Jasper Reports
Data Source	UNITS
Properties	*None*

Output

One sheet per unit in the CEOI, providing a complete summary of all data associated with that unit. Useful for pre-mission briefings or as a personal COMSEC card for individual operators.

> Copy `templates/jasper/UnitDetail.jrxml` to trim the report to only the fields relevant to your operations (e.g., remove asset sections if assets are distributed separately).

Related Topics

- Reports Overview
- Units
- Unit Editor
- Print Templates

Report: Group Roster



Prints a roster of all groups in the CEOI, listing each group's name and its member units. Useful as a network reference showing which units are assigned to each net.

Template Details

Field	Value
Template File	`GroupRoster.jrxml`
Template Type	Jasper Reports
Data Source	GROUPS
Properties	*None*

Output

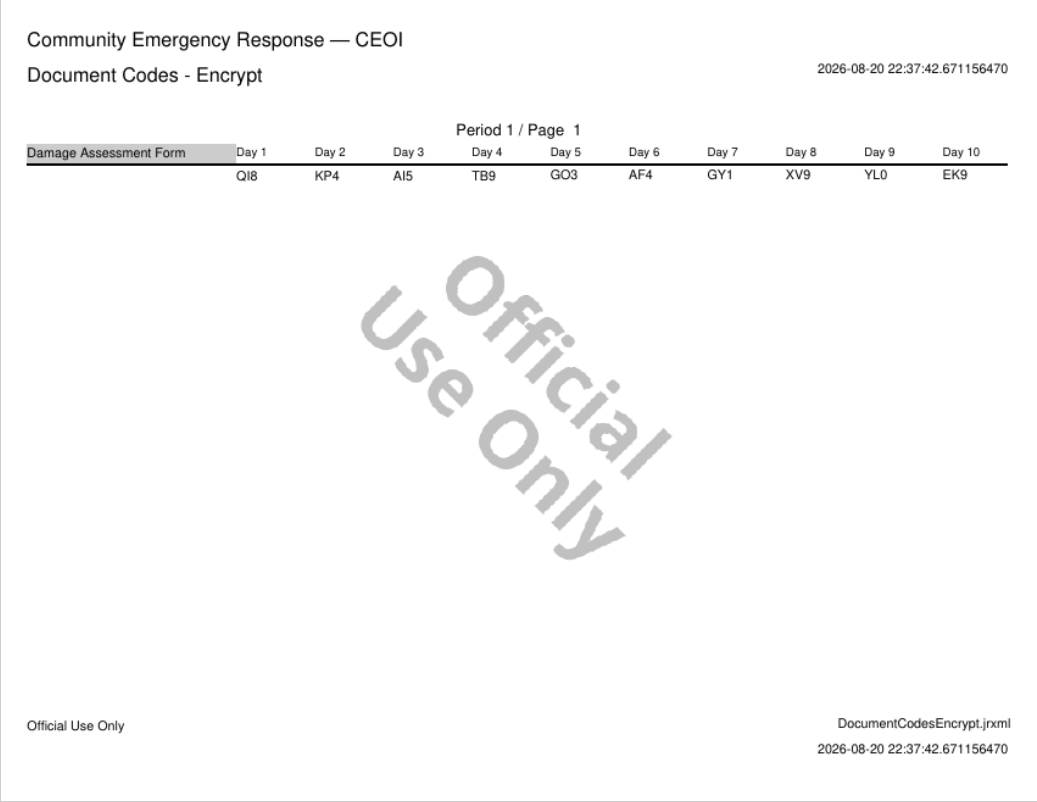
A report grouped by net/group, listing all member units under each group heading. The report covers all groups defined in the CEOI.

> Copy `templates/jasper/GroupRoster.jrxml` to add callsign columns or restructure the layout for a specific unit roster format.

Related Topics

- Reports Overview
- Groups
- Print Templates

Report: Document Codes – Encrypt


Community Emergency Response — CEOI

Document Codes - Encrypt

2026-08-20 22:37:42.671156470

Period 1 / Page 1

Damage Assessment Form	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
	QI8	KP4	AI5	TB9	GO3	AF4	GY1	XV9	YL0	EK9

Official Use Only

Official Use Only

DocumentCodesEncrypt.jrxml

2026-08-20 22:37:42.671156470

Prints a lookup table for encoding plain document names to their assigned codes. An operator who needs to reference a document over an unsecured channel looks up the plain name and transmits the code instead.

Template Details

Field	Value
Template File	`DocumentCodesEncrypt.jrxml`
Template Type	Jasper Reports
Data Source	DOCUMENT_ENCODE
Properties	*None*

Output

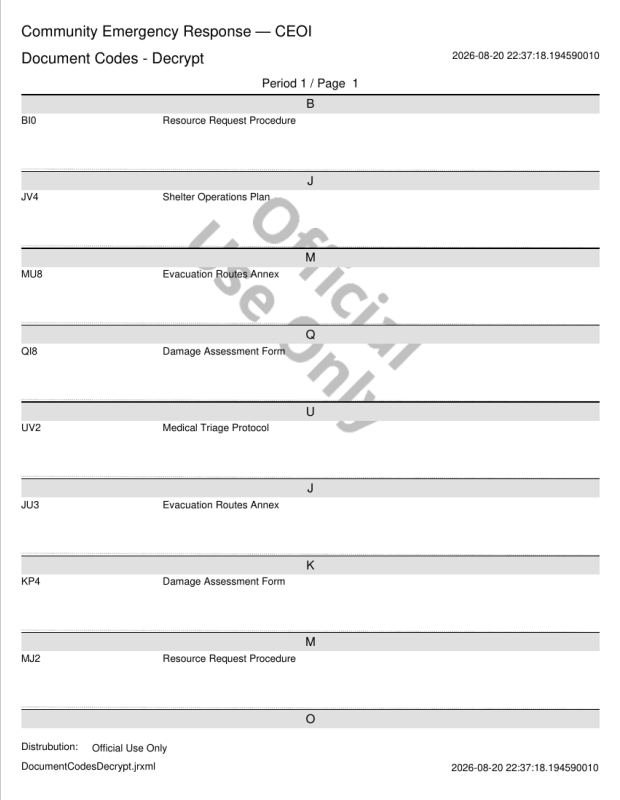
A two-column table: plain document name on the left, coded identifier on the right. Covers all documents in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Document Codes – Decrypt

- Documents
- Print Templates

Report: Document Codes – Decrypt


Community Emergency Response — CEOI
Document Codes - Decrypt
2026-08-20 22:37:18.194590010
Period 1 / Page 1
B
BI0 Resource Request Procedure
J
JV4 Shelter Operations Plan
M
MU8 Evacuation Routes Annex
Q
QI8 Damage Assessment Form
U
UV2 Medical Triage Protocol
J
JU3 Evacuation Routes Annex
K
KP4 Damage Assessment Form
M
MJ2 Resource Request Procedure
O
Distrubution: Official Use Only
DocumentCodesDecrypt.jrxml
2026-08-20 22:37:18.194590010
Official Use Only

Prints a lookup table for decoding document codes back to plain document names. An operator who receives a coded document reference can find the matching plain name in this table.

Template Details

Field	Value
Template File	`DocumentCodesDecrypt.jrxml`
Template Type	Jasper Reports
Data Source	DOCUMENT_DECODE
Properties	*None*

Output

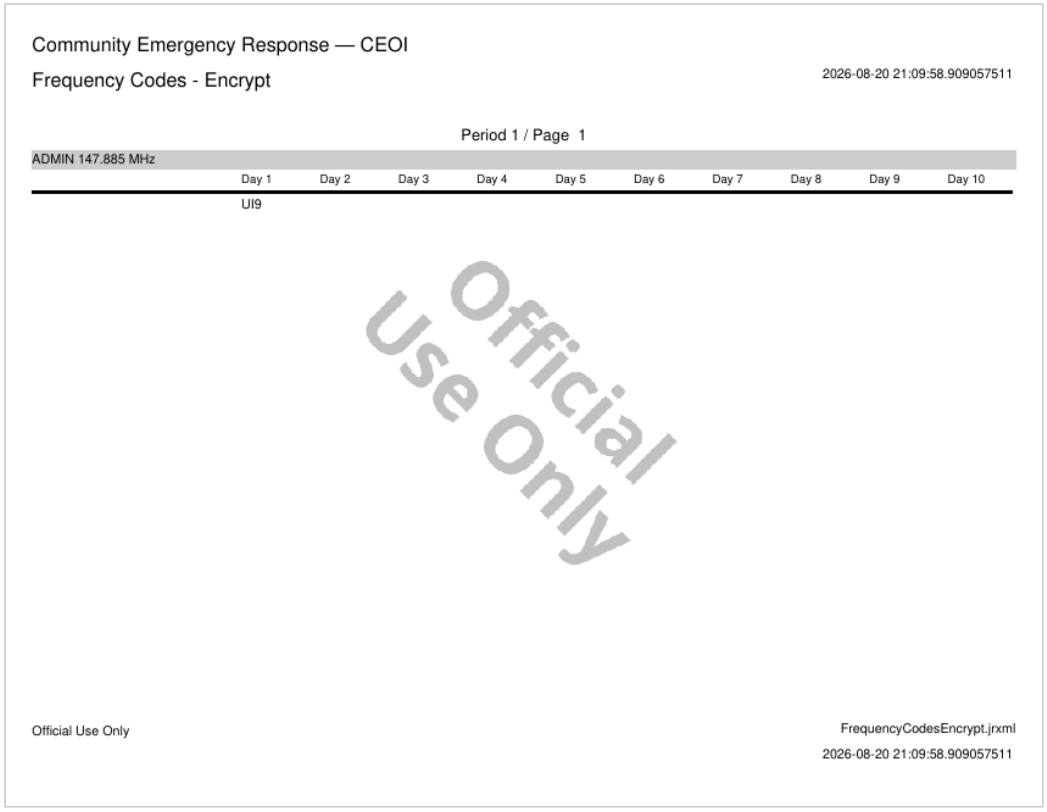
A two-column table: coded identifier on the left, plain document name on the right. Covers all documents in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Document Codes – Encrypt
- Documents

- Print Templates

Report: Frequency Codes – Encrypt



Prints a lookup table for encoding frequency values to their assigned codes. An operator who needs to transmit a frequency over an unsecured channel substitutes the code from this table.

Template Details

Field	Value
Template File	`FrequencyCodesEncrypt.jrxml`
Template Type	Jasper Reports
Data Source	FREQUENCY_ENCODE
Properties	*None*

Output

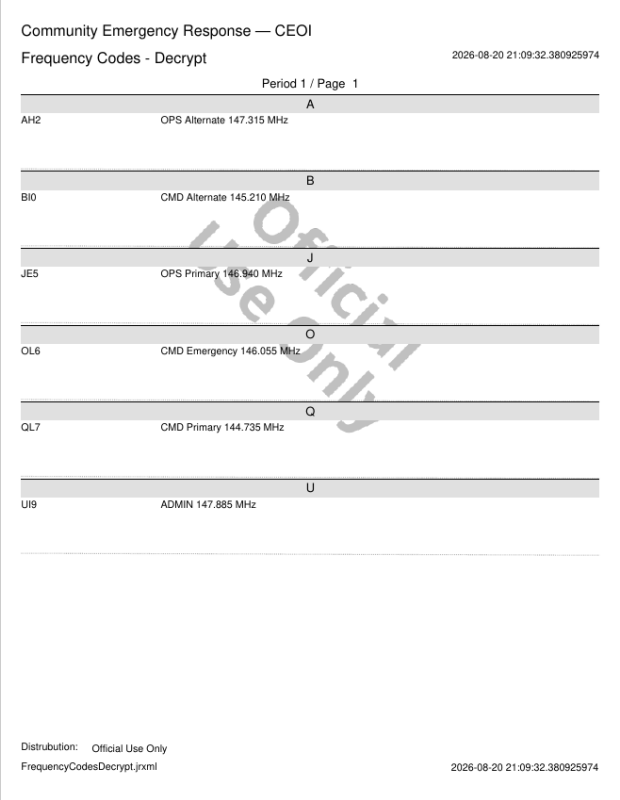
A two-column table: frequency value on the left, coded identifier on the right. Covers all frequencies in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Frequency Codes – Decrypt
- Frequencies

- Print Templates

Report: Frequency Codes – Decrypt


Community Emergency Response — CEOI
Frequency Codes - Decrypt
2026-08-20 21:09:32.380925974
Period 1 / Page 1
A
AH2 OPS Alternate 147.315 MHz
B
BI0 CMD Alternate 145.210 MHz
J
JE5 OPS Primary 146.940 MHz
O
OL6 CMD Emergency 146.055 MHz
Q
QL7 CMD Primary 144.735 MHz
U
UI9 ADMIN 147.885 MHz
Official Use Only
Distrubution: Official Use Only
FrequencyCodesDecrypt.jrxml
2026-08-20 21:09:32.380925974

Prints a lookup table for decoding frequency codes back to actual frequency values. An operator who receives a coded frequency reference looks up the code in this table to find the real frequency.

Template Details

Field	Value
Template File	`FrequencyCodesDecrypt.jrxml`
Template Type	Jasper Reports
Data Source	FREQUENCY_DECODE
Properties	*None*

Output

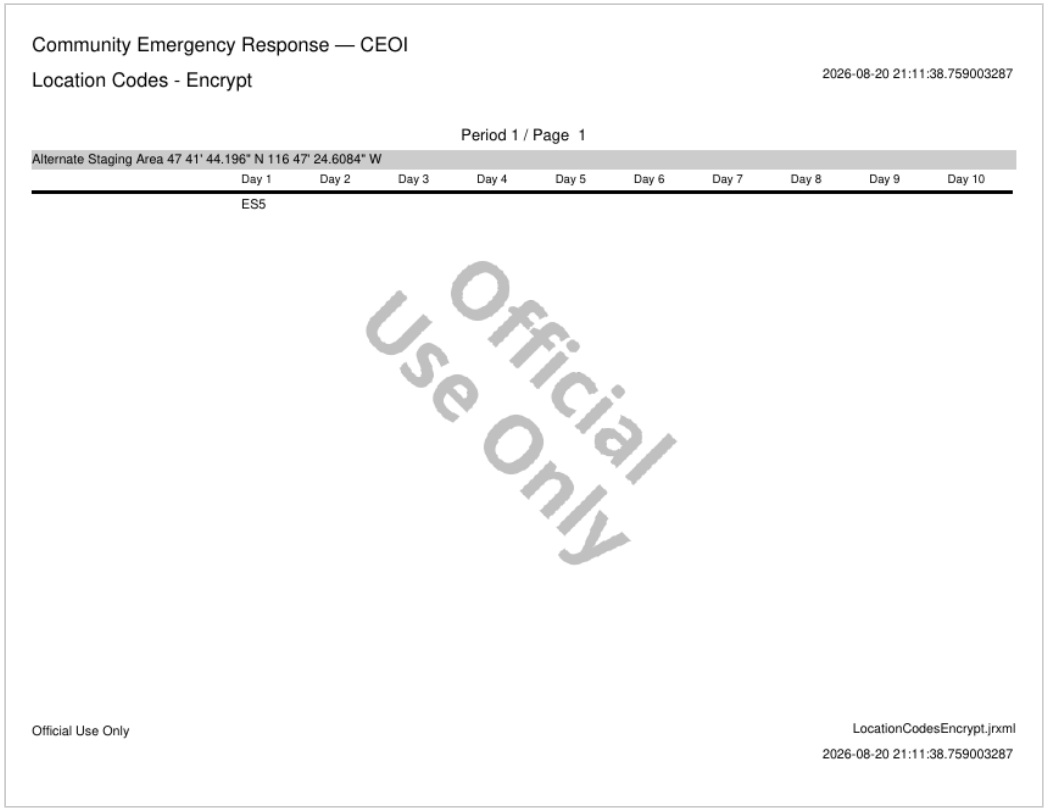
A two-column table: coded identifier on the left, frequency value on the right. Covers all frequencies in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Frequency Codes – Encrypt
- Frequencies

- Print Templates

Report: Location Codes – Encrypt



Prints a lookup table for encoding location coordinates to their assigned codes. An operator who needs to transmit a location over an unsecured channel substitutes the code for the real coordinates.

Template Details

Field	Value
Template File	`LocationCodesEncrypt.jrxml`
Template Type	Jasper Reports
Data Source	LOCATION_ENCODE
Properties	*None*

Output

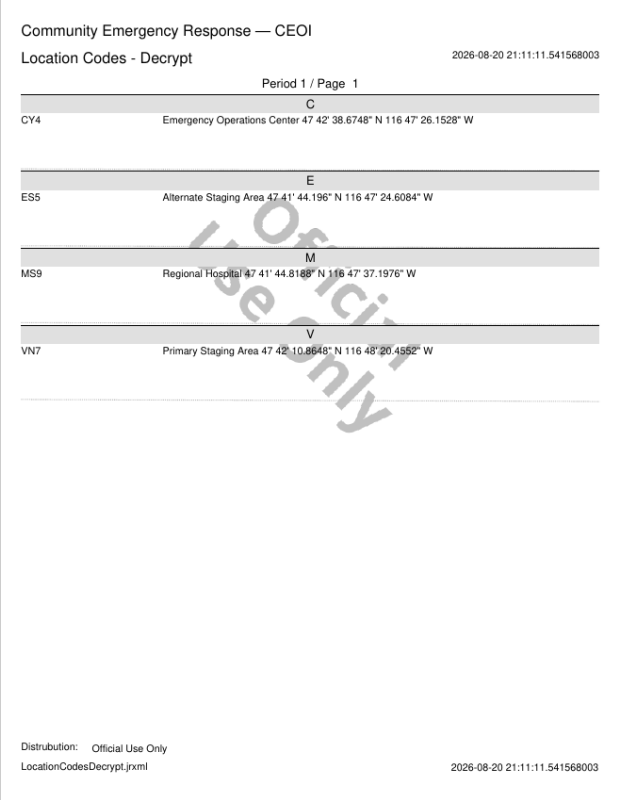
A two-column table: location name or coordinates on the left, coded identifier on the right. Covers all locations in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Location Codes – Decrypt
- Locations

- Print Templates

Report: Location Codes – Decrypt


Community Emergency Response — CEOI

Location Codes - Decrypt

2026-08-20 21:11:11.541568003

Period 1 / Page 1

C

CY4 Emergency Operations Center 47 42' 38.6748" N 116 47' 26.1528" W

E

ES5 Alternate Staging Area 47 41' 44.196" N 116 47' 24.6084" W

M

MS9 Regional Hospital 47 41' 44.8188" N 116 47' 37.1976" W

V

VN7 Primary Staging Area 47 42' 10.8648" N 116 48' 20.4552" W

Official Use Only

Distrubution: Official Use Only

LocationCodesDecrypt.jrxml

2026-08-20 21:11:11.541568003

Prints a lookup table for decoding location codes back to actual coordinates. An operator who receives a coded location reference looks up the code to find the real grid or coordinate value.

Template Details

Field	Value
Template File	`LocationCodesDecrypt.jrxml`
Template Type	Jasper Reports
Data Source	LOCATION_DECODE
Properties	*None*

Output

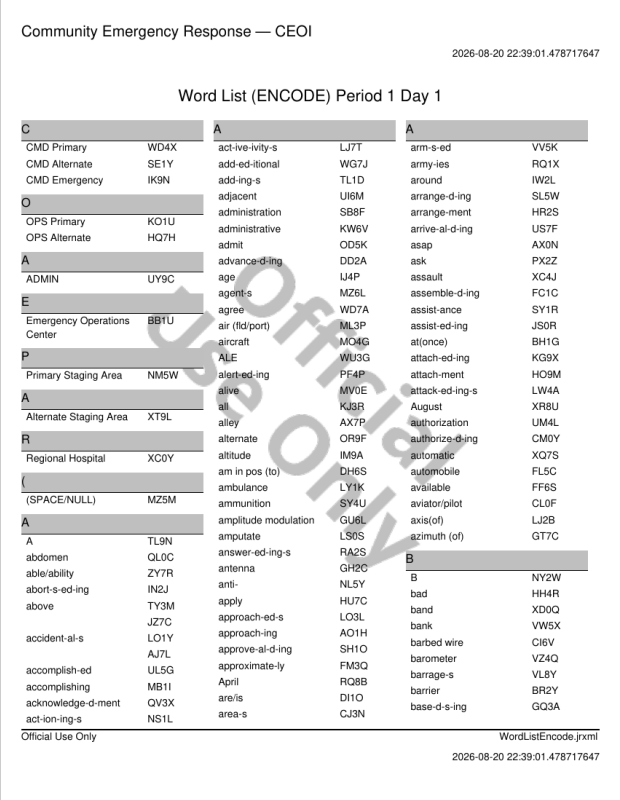
A two-column table: coded identifier on the left, location coordinates on the right. Covers all locations in the CEOI across all periods and divisions.

Related Topics

- Reports Overview
- Location Codes – Encrypt
- Locations

- Print Templates

Report: Wordlist Encode



Community Emergency Response — CEOI

2026-08-20 22:39:01.478717647

Word List (ENCODE) Period 1 Day 1

C	
CMD Primary	WD4X
CMD Alternate	SE1Y
CMD Emergency	IK9N
O	
OPS Primary	KO1U
OPS Alternate	HQ7H
A	
ADMIN	UY9C
E	
Emergency Operations Center	BB1U
P	
Primary Staging Area	NM5W
A	
Alternate Staging Area	XT9L
R	
Regional Hospital	XC0Y
(	
(SPACE/NULL)	MZ5M
A	
A	TL9N
abdomen	QL0C
able/ability	ZY7R
abort-s-ed-ing	IN2J
above	TY3M
	JZ7C
accident-al-s	LO1Y
	AJ7L
accomplish-ed	UL5G
accomplishing	MB1I
acknowledge-d-ment	QV3X
act-ion-ing-s	NS1L

A	
act-ive-ivity-s	LJ7T
add-ed-itional	WG7J
add-ing-s	TL1D
adjacent	UI6M
administration	SB8F
administrative	KW6V
admit	OD5K
advance-d-ing	DD2A
age	IJ4P
agent-s	MZ6L
agree	WD7A
air (fld/port)	ML3P
aircraft	MO4G
ALE	WU3G
alert-ed-ing	PF4P
alive	MV0E
all	KJ3R
alley	AX7P
alternate	OR9F
altitude	IM9A
am in pos (to)	DH6S
ambulance	LY1K
ammunition	SY4U
amplitude modulation	GU6L
amputate	LS0S
answer-ed-ing-s	RA2S
antenna	GH2C
anti-	NL5Y
apply	HU7C
approach-ed-s	LO3L
approach-ing	AO1H
approve-al-d-ing	SH1O
approximate-ly	FM3Q
April	RQ8B
are/is	DI1O
area-s	CJ3N

A	
arm-s-ed	VV5K
army-ies	RQ1X
around	IW2L
arrange-d-ing	SL5W
arrange-ment	HR2S
arrive-al-d-ing	US7F
asap	AX0N
ask	PX2Z
assault	XC4J
assemble-d-ing	FC1C
assist-ance	SY1R
assist-ed-ing	JS0R
at(once)	BH1G
attach-ed-ing	KG9X
attach-ment	HO9M
attack-ed-ing-s	LW4A
August	XR8U
authorization	UM4L
authorize-d-ing	CM0Y
automatic	XQ7S
automobile	FL5C
available	FF6S
aviator/pilot	CL0F
axis(of)	LJ2B
azimuth (of)	GT7C
B	
B	NY2W
bad	HH4R
band	XD0Q
bank	VW5X
barbed wire	CI6V
barometer	VZ4Q
barrage-s	VL8Y
barrier	BR2Y
base-d-s-ing	GQ3A

Official Use Only

WordListEncode.jrxml

2026-08-20 22:39:01.478717647

Prints a lookup table for encoding plain words or phrases to their assigned codes. An operator who needs to refer to a word list item over an unsecured channel substitutes the code for the plain word.

Template Details

Field	Value
Template File	`WordListEncode.jrxml`
Template Type	Jasper Reports
Data Source	WORDLIST_ENCODE
Properties	*None*

Output

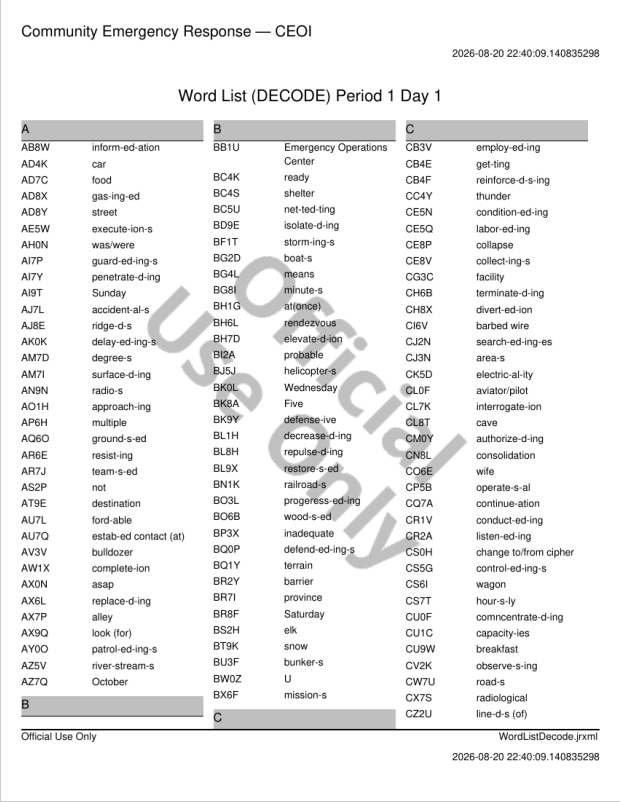
A two-column table: plain word or phrase on the left, coded identifier on the right. Covers the entire word list in the CEOI, sorted for easy lookup by plaintext.

Related Topics

- Reports Overview
- Wordlist Decode
- Word List

- Word List Encoder/Decoder
- Print Templates

Report: Wordlist Decode



Community Emergency Response — CEOI

2026-08-20 22:40:09.140835298

Word List (DECODE) Period 1 Day 1

A		B		C	
AB8W	inform-ed-ation	BB1U	Emergency Operations Center	CB3V	employ-ed-ing
AD4K	car	BC4K	ready	CB4E	get-ting
AD7C	food	BC4S	shelter	CB4F	reinforce-d-s-ing
AD8X	gas-ing-ed	BC5U	net-ted-ting	CC4Y	thunder
AD8Y	street	BD9E	isolate-d-ing	CE5N	condition-ed-ing
AE5W	execute-ion-s	BF1T	storm-ing-s	CE5Q	labor-ed-ing
AH0N	was/were	BG2D	boat-s	CE8P	collapse
AI7P	guard-ed-ing-s	BG4L	means	CE8V	collect-ing-s
AI7Y	penetrate-d-ing	BG8I	minute-s	CG3C	facility
AI9T	Sunday	BH1G	at(once)	CH6B	terminate-d-ing
AJ7L	accident-al-s	BH6L	rendezvous	CH8X	divert-ed-ion
AJ8E	ridge-d-s	BH7D	elevate-d-ion	CI6V	barbed wire
AK0K	delay-ed-ing-s	BI2A	probable	CJ2N	search-ed-ing-es
AM7D	degree-s	BJ5J	helicopter-s	CJ3N	area-s
AM7I	surface-d-ing	BK0L	Wednesday	CK5D	electric-al-ity
AN9N	radio-s	BK8A	Five	CL0F	aviator/pilot
AO1H	approach-ing	BK9Y	defense-ive	CL7K	interrogate-ion
AP6H	multiple	BL1H	decrease-d-ing	CL8T	cave
AQ6O	ground-s-ed	BL8H	repulse-d-ing	CM0Y	authorize-d-ing
AR6E	resist-ing	BL9X	restore-s-ed	CN8L	consolidation
AR7J	team-s-ed	BN1K	railroad-s	CQ6E	wife
AS2P	not	BO3L	progeress-ed-ing	CP5B	operate-s-al
AT9E	destination	BO6B	wood-s-ed	CQ7A	continue-ation
AU7L	ford-able	BP3X	inadequate	CR1V	conduct-ed-ing
AU7Q	estab-ed contact (at)	BQ0P	defend-ed-ing-s	CR2A	listen-ed-ing
AV3V	bulldozer	BQ1Y	terrain	CS0H	change to/from cipher
AW1X	complete-ion	BR2Y	barrier	CS5G	control-ed-ing-s
AX0N	asap	BR7I	province	CS6I	wagon
AX6L	replace-d-ing	BR8F	Saturday	CS7T	hour-s-ly
AX7P	alley	BS2H	elk	CU0F	comncentrate-d-ing
AX9Q	look (for)	BT9K	snow	CU1C	capacity-ies
AY0O	patrol-ed-ing-s	BU3F	bunker-s	CU9W	breakfast
AZ5V	river-stream-s	BW0Z	U	CV2K	observe-s-ing
AZ7Q	October	BX6F	mission-s	CW7U	road-s
B		C		CX7S	radiological
				CZ2U	line-d-s (of)

Official Use Only

Official Use Only

WordListDecode.jrxml

2026-08-20 22:40:09.140835298

Prints a lookup table for decoding word codes back to their plain meaning. An operator who receives a coded word reference looks up the code in this table to find the original word or phrase.

Template Details

Field	Value
Template File	`WordListDecode.jrxml`
Template Type	Jasper Reports
Data Source	WORDLIST_DECODE
Properties	*None*

Output

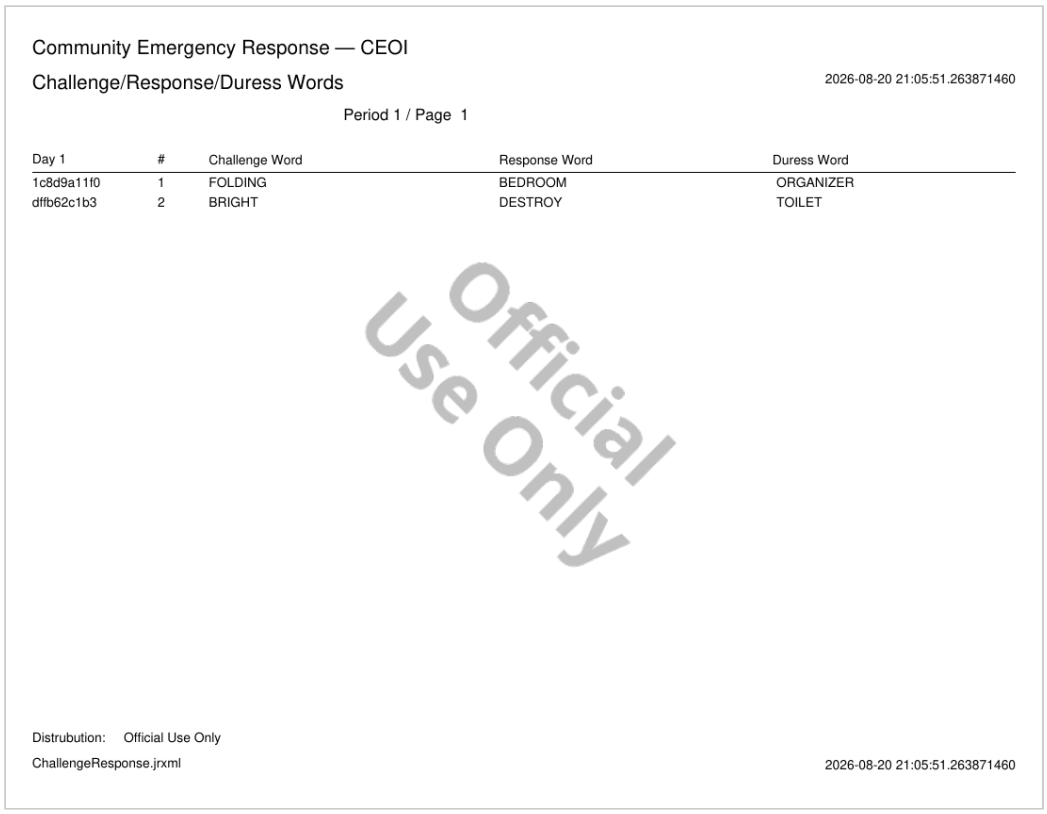
A two-column table: coded identifier on the left, plain word or phrase on the right. Covers the entire word list in the CEOI.

Related Topics

- Reports Overview
- Wordlist Encode
- Word List

- Word List Encoder/Decoder
- Print Templates

Report: Challenge Response Words


Community Emergency Response — CEOI

Challenge/Response/Duress Words

2026-08-20 21:05:51.263871460

Period 1 / Page 1

Day 1	#	Challenge Word	Response Word	Duress Word
1c8d9a11f0	1	FOLDING	BEDROOM	ORGANIZER
dffb62c1b3	2	BRIGHT	DESTROY	TOILET

Distrubution: Official Use Only

ChallengeResponse.jrxml

2026-08-20 21:05:51.263871460

Prints the challenge/response/duress word pairs from the CEOI. Each row shows the challenge word, the correct response word, and the duress word for one entry, grouped by period and division.

Template Details

Field	Value
Template File	`ChallengeResponse.jrxml`
Template Type	Jasper Reports
Data Source	CHALLENGE_RESPONSE
Properties	*None*

Output

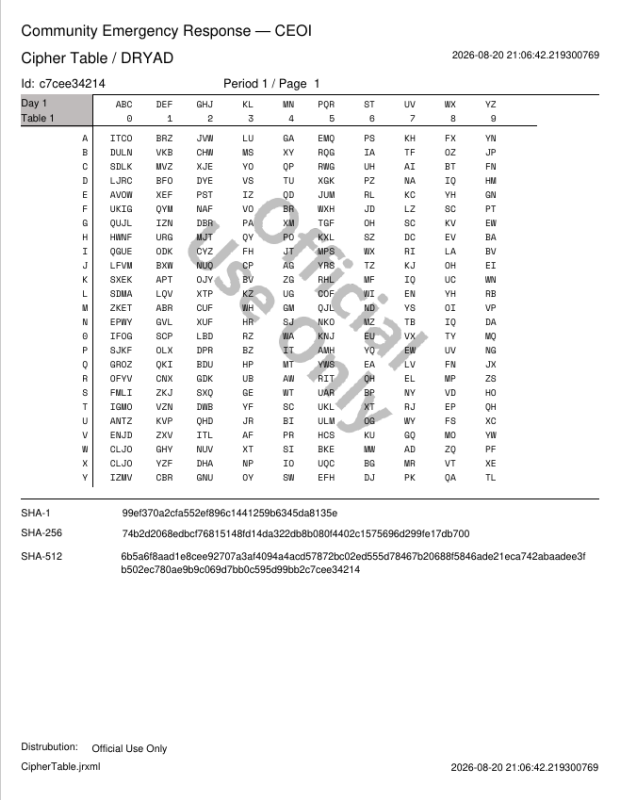
A table of all challenge/response pairs in the CEOI, covering all periods and divisions. The three words for each entry are printed side by side to make field use straightforward.

> Copy `templates/jasper/ChallengeResponse.jrxml` to adjust column layout or add a classification banner.

Related Topics

- Reports Overview
- Challenge/Response
- Print Templates

Report: DRYADs



Community Emergency Response — CEOI

Cipher Table / DRYAD

2026-08-20 21:06:42.219300769

Id: c7cee34214 Period 1 / Page 1

Day 1 Table 1	ABC 0	DEF 1	GHJ 2	KL 3	MN 4	PQR 5	ST 6	UV 7	WX 8	YZ 9
A	ITCO	BRZ	JVW	LU	GA	EMQ	PS	KH	FX	YN
B	DULN	VKB	CHW	MS	XY	RQG	IA	TF	OZ	JP
C	SDLK	MVZ	XJE	YO	QP	RWG	UH	AI	BT	FN
D	LJRC	BFO	DYE	VS	TU	XGK	PZ	NA	IQ	HM
E	AVOW	XEF	PST	IZ	QD	JUM	RL	KC	YH	GN
F	UKIG	QVM	NAF	VO	BR	WXH	JD	LZ	SC	PT
G	QUJL	IZN	DBR	PA	XM	TGF	OH	SC	KV	EW
H	HWNF	URG	MJT	QY	PO	KXL	SZ	DC	EV	BA
I	QGUE	ODK	CYZ	FH	JT	MPS	WX	RI	LA	BV
J	LFVM	BXW	NUQ	CP	AG	YRB	TZ	KJ	OH	EI
K	SXEK	APT	OJY	BV	ZG	RHL	MF	IQ	UC	WN
L	SDWA	LQV	XTP	KZ	UG	COF	WI	EN	YH	RB
M	ZKET	ABR	CUF	WH	GW	QJL	ND	YS	OI	VP
N	EPWY	GVL	XUF	HR	SJ	NKO	MZ	TB	IQ	DA
O	IFOG	SCP	LBD	RZ	WA	KNJ	EU	VX	TY	WQ
P	SJKF	OLX	DPR	BZ	IT	AMH	YQ	EW	UV	NG
Q	GROZ	QKI	BDU	HP	MT	YWB	EA	LV	FN	JX
R	OFYV	CNX	GDK	UB	AW	RIT	QH	EL	MP	ZS
S	FMLI	ZKJ	SXQ	GE	WT	UAR	BP	NY	VD	HO
T	IGWO	VZN	DWB	YF	SC	UKL	XT	RJ	EP	QH
U	ANTZ	KVP	QHD	JR	BI	ULM	OG	WY	FS	XC
V	ENJD	ZXV	ITL	AF	PR	HCS	KU	GQ	MO	YW
W	CLJO	GHY	NUV	XT	SI	BKE	MW	AD	ZQ	PF
X	CLJO	YZF	DHA	NP	IO	UQC	BG	MR	VT	XE
Y	IZMV	CBR	GNU	OY	SW	EFH	DJ	PK	QA	TL

SHA-1 99ef370a2cfa552ef896c1441259b6345da8135e

SHA-256 74b2d2068edbcf76815148fd14da322db8b080f4402c1575696d299fe17db700

SHA-512 6b5a6f8aad1e8cee92707a3af4094a4acd57872bc02ed555d78467b20688f5846ade21eca742abaadee3fb502ec780ae9b9c069d7bb0c595d99bb2c7cee34214

Distrubution: Official Use Only

CipherTable.jrxml

2026-08-20 21:06:42.219300769

Official Use Only

Prints DRYADs. Each DRYAD table occupies its own section, showing the grid of letter columns and digit rows used to encrypt and decrypt numerical values such as grid coordinates, frequencies, and times.

Template Details

Field	Value
Template File	`CipherTable.jrxml`
Template Type	Jasper Reports
Data Source	CYPHERTABLES
Properties	*None*

Output

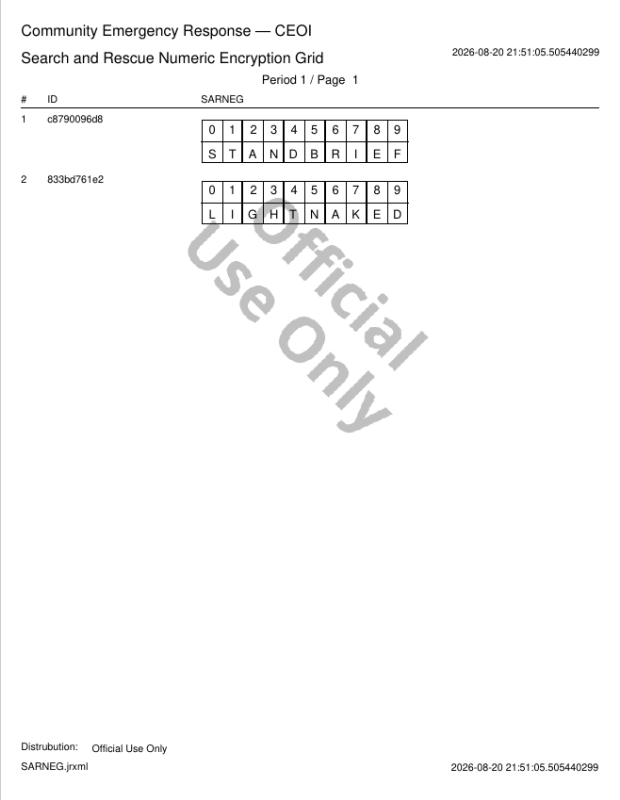
One section per DRYAD table defined in the CEOI, covering all periods and divisions. The grid layout matches the standard DRYAD format for field use.

> Copy `templates/jasper/CipherTable.jrxml` to add a classification banner, change the grid font size, or print multiple tables per page.

Related Topics

- Reports Overview
- DRYADs
- DRYAD Viewer
- Print Templates

Report: SARNEGs



Prints SARNEG (Search and Rescue Numerical Encryption Grid) tables. Each SARNEG occupies its own section, showing the grid used to encode numerical values during search and rescue operations.

Template Details

Field	Value
Template File	`SARNEG.jrxml`
Template Type	Jasper Reports
Data Source	SARNEG
Properties	*None*

Output

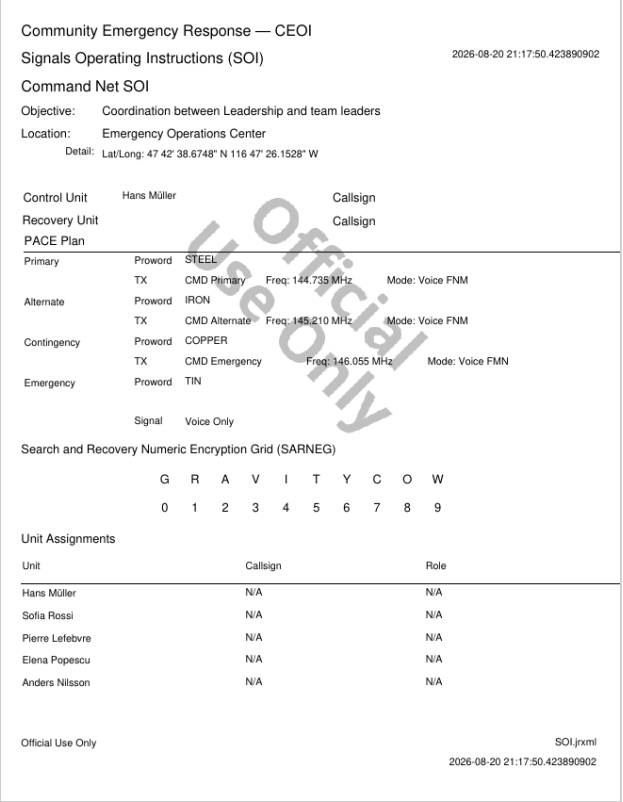
One section per SARNEG defined in the CEOI, covering all periods and divisions. The grid layout matches the standard SARNEG field format.

> Copy `templates/jasper/SARNEG.jrxml` to add a classification banner or change the grid font for high-contrast printing.

Related Topics

- Reports Overview
- SARNEGs
- SARNEG Viewer
- Print Templates

Report: SOI



Community Emergency Response — CEOI

Signals Operating Instructions (SOI) 2026-08-20 21:17:50.423890902

Command Net SOI

Objective: Coordination between Leadership and team leaders

Location: Emergency Operations Center

Detail: Lat/Long: 47 42' 38.6748" N 116 47' 26.1528" W

Control Unit Hans Müller Callsign

Recovery Unit Callsign

PACE Plan

Primary	Proword	STEEL			
	TX	CMD Primary	Freq: 144.735 MHz	Mode: Voice FNM	
Alternate	Proword	IRON			
	TX	CMD Alternate	Freq: 145.210 MHz	Mode: Voice FNM	
Contingency	Proword	COPPER			
	TX	CMD Emergency	Freq: 146.055 MHz	Mode: Voice FMN	
Emergency	Proword	TIN			
	Signal	Voice Only			

Official Use Only

Search and Recovery Numeric Encryption Grid (SARNEG)

G	R	A	V	I	T	Y	C	O	W
0	1	2	3	4	5	6	7	8	9

Unit Assignments

Unit	Callsign	Role
Hans Müller	N/A	N/A
Sofia Rossi	N/A	N/A
Pierre Lefebvre	N/A	N/A
Elena Popescu	N/A	N/A
Anders Nilsson	N/A	N/A

Official Use Only SOI.jrxml

2026-08-20 21:17:50.423890902

Prints a Signals Operating Instruction sheet for each SOI defined in the CEOI. Each sheet includes the SOI details, assigned frequencies, PACE plan, communication schedule, and unit assignments.

Template Details

Field	Value
Template File	`SOI.jrxml`
Template Type	Jasper Reports
Data Source	SOI
Properties	*None*

Output

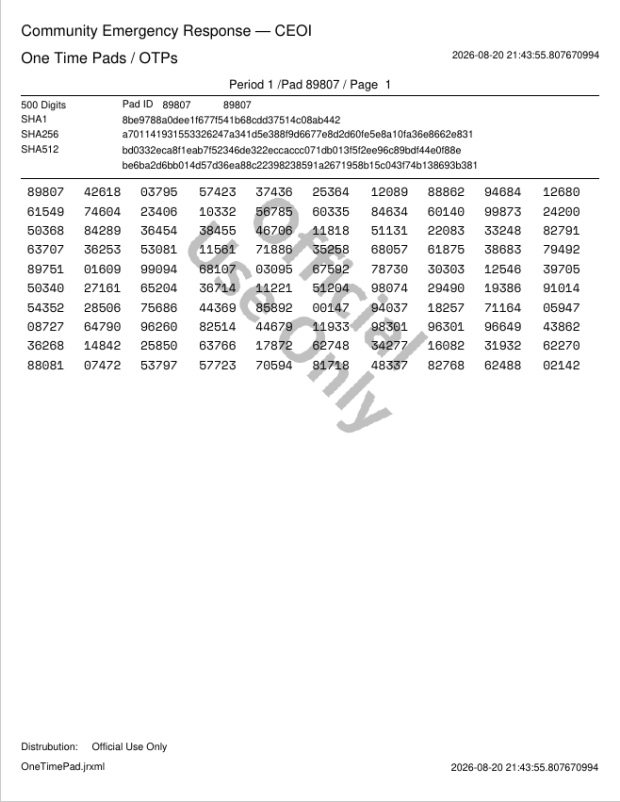
One or more pages per SOI, covering primary and alternate frequencies, the PACE plan, the communication schedule, and the list of assigned units. Designed to be distributed to net members as a self-contained reference card.

> Copy `templates/jasper/SOI.jrxml` to reorganize the layout, add sub-reports, or change the frequency table columns.

Related Topics

- Reports Overview
- Signals Operating Instructions
- SOI Editor
- Print Templates

Report: One Time Pad



Community Emergency Response — CEOI

One Time Pads / OTPs

2026-08-20 21:43:55.807670994

Period 1 /Pad 89807 / Page 1

500 Digits	Pad ID 89807	89807
SHA1	8be9788a0dee1f677f541b68cdd37514c08ab442	
SHA256	a7011419315533262 47a341d5e388f9d6677e8d2d60fe5e8a10fa36e8662e8831	
SHA512	bd0332eca8f1eab7f52346de322eccaccc071db013f5f2ee96c89bdf44e0f88e be6ba2d6bb014d57d36ea88c22398238591a2671958b15c043f74b138693b381	

89807	42618	03795	57423	37436	25364	12089	88862	94684	12680
61549	74604	23406	10332	56785	60335	84634	60140	99873	24200
50368	84289	36454	38455	46706	11818	51131	22083	33248	82791
63707	36253	53081	11561	71888	35258	68057	61875	38683	79492
89751	01609	99094	68107	03095	67592	78730	30303	12546	39705
50340	27161	65204	36714	11221	51204	98074	29490	19386	91014
54352	28506	75686	44369	85892	00147	94037	18257	71164	05947
08727	64790	96260	82514	44679	11933	98301	96301	96649	43862
36268	14842	25850	63766	17872	62748	34277	16082	31932	62270
88081	07472	53797	57723	70594	81718	48337	82768	62488	02142

Official Use Only

Distrubution: Official Use Only

OneTimePad.jrxml

2026-08-20 21:43:55.807670994

Prints a standard one-time pad sheet. Random values are laid out in groups across the page. This is the baseline OTP layout without directional labeling; use it when pads are not separated into inbound and outbound sets.

Template Details

Field	Value
Template File	`OneTimePad.jrxml`
Template Type	Jasper Reports
Data Source	ONE_TIME_PAD
Properties	`groupcount=10`

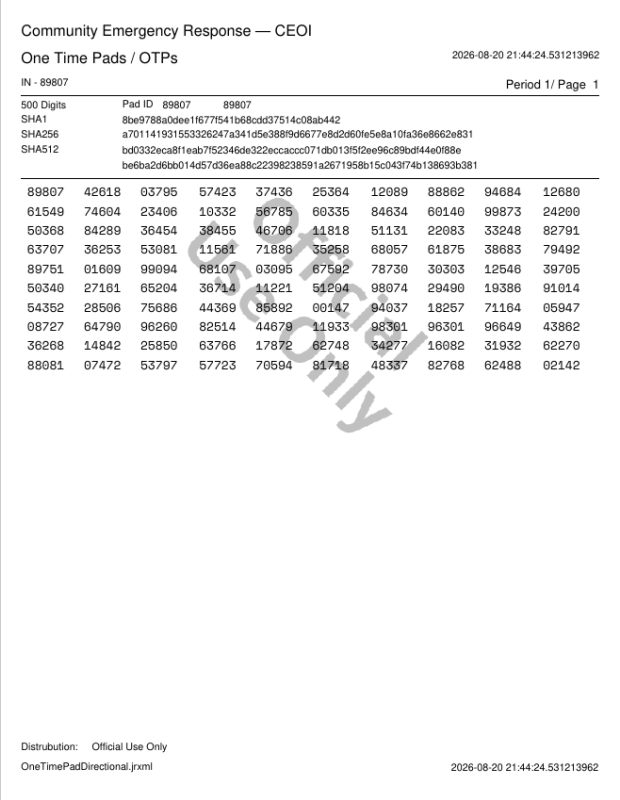
Properties

Property	Description
`groupcount`	Number of groups printed per line. Default is `10`. Reduce for smaller paper or larger fonts; increase to fit more data per line.

Related Topics

- Reports Overview
- One Time Pad IN
- One Time Pad OUT
- One Time Pads
- Print Templates

Report: One Time Pad IN



Community Emergency Response — CEOI

One Time Pads / OTPs

2026-08-20 21:44:24.531213962

IN - 89807 Period 1/ Page 1

500 Digits	Pad ID 89807	89807
SHA1	8be9788a0dee1f677f541b68cdd37514c08ab442	
SHA256	a701141931553326247a341d5e388f9d6677e8d2d60fe5e8a10fa36e8662e831	
SHA512	bd0332eca8f1eab7f52346de322eccaccc071db013f5f2ee96c89bdf44e0f88e be6ba2d6bb014d57d36ea88c22398238591a2671958b15c043f74b138693b381	

89807	42618	03795	57423	37436	25364	12089	88862	94684	12680
61549	74604	23406	10332	56785	60335	84634	60140	99873	24200
50368	84289	36454	38455	46706	11818	51131	22083	33248	82791
63707	36253	53081	11561	71888	35258	68057	61875	38683	79492
89751	01609	99094	68107	03095	67592	78730	30303	12546	39705
50340	27161	65204	36714	11221	51204	98074	29490	19386	91014
54352	28506	75686	44369	85892	00147	94037	18257	71164	05947
08727	64790	96260	82514	44679	11933	98301	96301	96649	43862
36268	14842	25850	63766	17872	62748	34277	16082	31932	62270
88081	07472	53797	57723	70594	81718	48337	82768	62488	02142

Official Use Only

Distrubution: Official Use Only

OneTimePadDirectional.jrxml

2026-08-20 21:44:24.531213962

Prints a directional one-time pad sheet labeled for incoming messages. Use this when pads are split into separate inbound and outbound sets so operators can quickly identify the correct pad for a received transmission.

Template Details

Field	Value
Template File	`OneTimePadDirectional.jrxml`
Template Type	Jasper Reports
Data Source	ONE_TIME_PAD
Properties	`direction=IN` `groupcount=10`

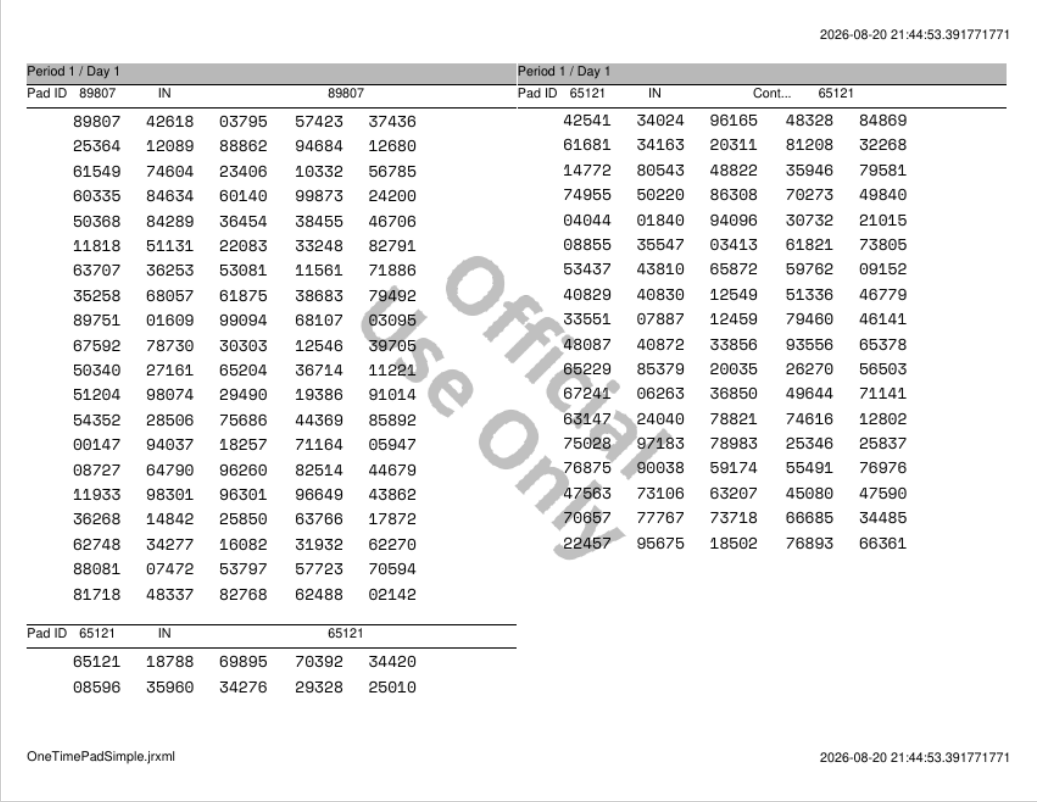
Properties

Property	Description
`direction`	Sets the directional label on the pad header. `IN` marks pads for incoming traffic. Use `OUT` for outgoing (see One Time Pad OUT).
`groupcount`	Number of groups per line. Default is `10`.

Related Topics

- Reports Overview
- One Time Pad OUT
- One Time Pad IN Simplified
- One Time Pad
- One Time Pads
- Print Templates

Report: One Time Pad IN Simplified



2026-08-20 21:44:53.391771771

Period 1 / Day 1

Pad ID	89807	IN			89807
	89807	42618	03795	57423	37436
	25364	12089	88862	94684	12680
	61549	74604	23406	10332	56785
	60335	84634	60140	99873	24200
	50368	84289	36454	38455	46706
	11818	51131	22083	33248	82791
	63707	36253	53081	11561	71886
	35258	68057	61875	38683	79492
	89751	01609	99094	68107	03095
	67592	78730	30303	12546	39705
	50340	27161	65204	36714	11221
	51204	98074	29490	19386	91014
	54352	28506	75686	44369	85892
	00147	94037	18257	71164	05947
	08727	64790	96260	82514	44679
	11933	98301	96301	96649	43862
	36268	14842	25850	63766	17872
	62748	34277	16082	31932	62270
	88081	07472	53797	57723	70594
	81718	48337	82768	62488	02142

Period 1 / Day 1

Pad ID	65121	IN	Cont...	65121	
	42541	34024	96165	48328	84869
	61681	34163	20311	81208	32268
	14772	80543	48822	35946	79581
	74955	50220	86308	70273	49840
	04044	01840	94096	30732	21015
	08855	35547	03413	61821	73805
	53437	43810	65872	59762	09152
	40829	40830	12549	51336	46779
	33551	07887	12459	79460	46141
	48087	40872	33856	93556	65378
	65229	85379	20035	26270	56503
	67241	06263	36850	49644	71141
	63147	24040	78821	74616	12802
	75028	97183	78983	25346	25837
	76875	90038	59174	55491	76976
	47563	73106	63207	45080	47590
	70057	77767	73718	66685	34485
	22457	95675	18502	76893	66361

Pad ID	65121	IN			65121
	65121	18788	69895	70392	34420
	08596	35960	34276	29328	25010

Official Use Only

OneTimePadSimple.jrxml

2026-08-20 21:44:53.391771771

Prints a simplified incoming one-time pad with 5 groups per line instead of 10. The wider spacing makes individual groups easier to read in the field and reduces transcription errors.

Template Details

Field	Value
Template File	`OneTimePadSimple.jrxml`
Template Type	Jasper Reports
Data Source	ONE_TIME_PAD
Properties	`groupcount=5` `direction=IN`

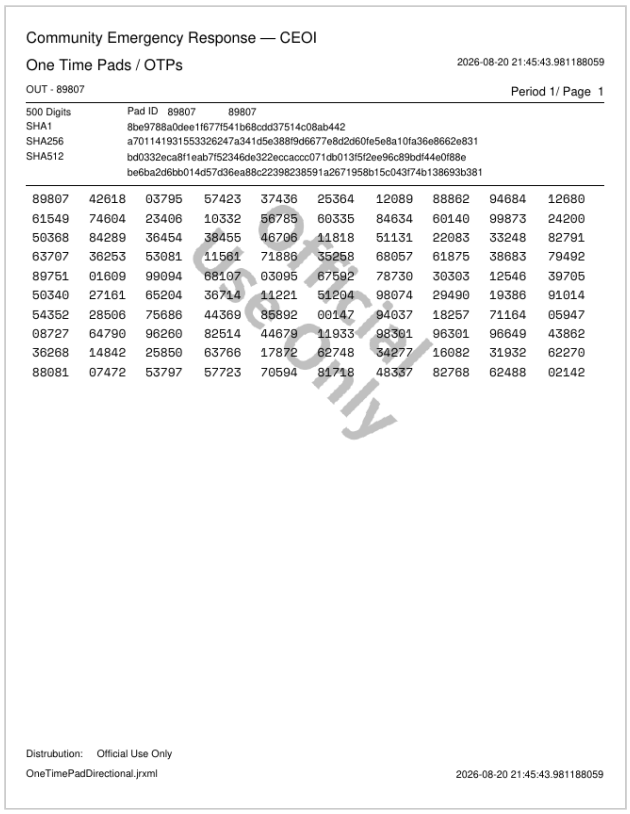
Properties

Property	Description
`groupcount`	Number of groups per line. Default is `5` for this template.
`direction`	Directional label on the pad header (`IN` or `OUT`).

Related Topics

- Reports Overview
- One Time Pad IN
- One Time Pad OUT Simplified
- One Time Pads
- Print Templates

Report: One Time Pad OUT



Community Emergency Response — CEOI

One Time Pads / OTPs

2026-08-20 21:45:43.981188059

OUT - 89807 Period 1/ Page 1

500 Digits Pad ID 89807 89807

SHA1 8be9788a0dee1f677f541b68cdd37514c08ab442

SHA256 a701141931553326247a341d5e388f9d6677e8d2d60fe5e8a10fa36e8662e831

SHA512 bd0332eca8f1eab7f52346de322eccaccc071db013f5f2ee96c89bdf44e0f88e
be6ba2d6bb014d57d36ea88c22398238591a2671958b15c043f74b138693b381

89807	42618	03795	57423	37436	25364	12089	88862	94684	12680
61549	74604	23406	10332	56785	60335	84634	60140	99873	24200
50368	84289	36454	38455	46706	11818	51131	22083	33248	82791
63707	36253	53081	11561	71888	35258	68057	61875	38683	79492
89751	01609	99094	68107	03095	67592	78730	30303	12546	39705
50340	27161	65204	36714	11221	51204	98074	29490	19386	91014
54352	28506	75686	44369	85892	00147	94037	18257	71164	05947
08727	64790	96260	82514	44679	11933	98301	96301	96649	43862
36268	14842	25850	63766	17872	62748	34277	16082	31932	62270
88081	07472	53797	57723	70594	81718	48337	82768	62488	02142

Official Use Only

Distrubution: Official Use Only

OneTimePadDirectional.jrxml

2026-08-20 21:45:43.981188059

Prints a directional one-time pad sheet labeled for outgoing messages. Use this when pads are split into separate inbound and outbound sets so operators can quickly locate the correct pad before transmitting.

Template Details

Field	Value
Template File	`OneTimePadDirectional.jrxml`
Template Type	Jasper Reports
Data Source	ONE_TIME_PAD
Properties	`direction=OUT` `groupcount=10`

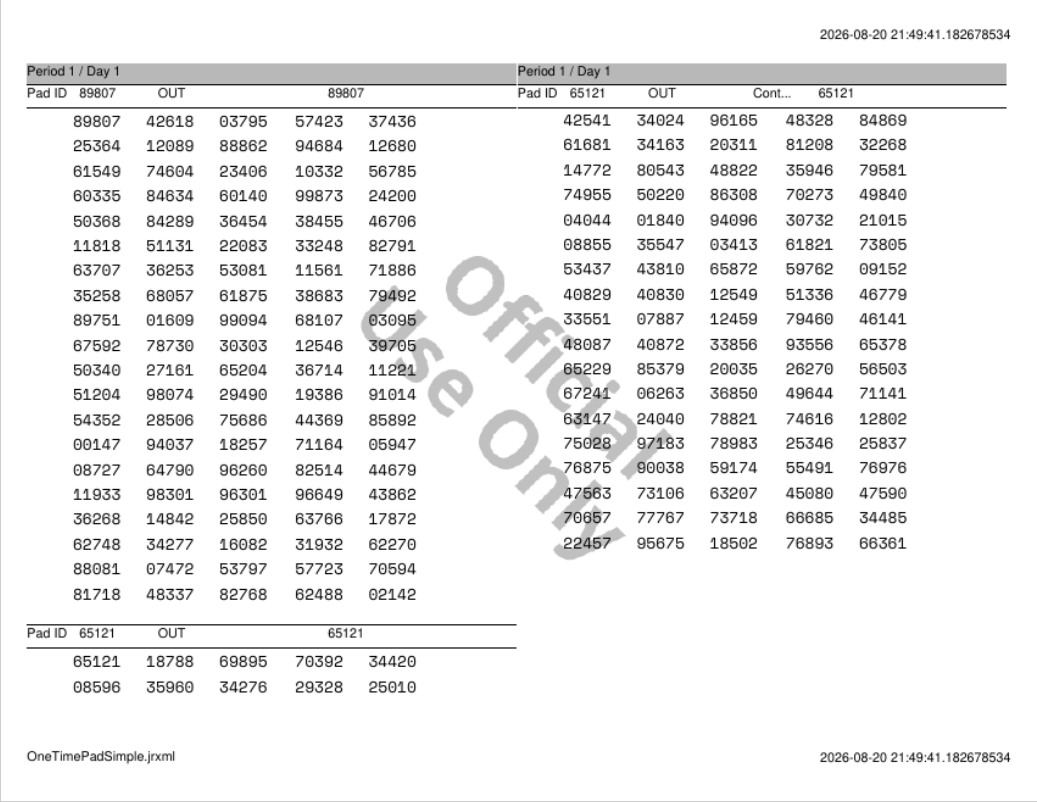
Properties

Property	Description
`direction`	Sets the directional label. `OUT` marks pads for outgoing traffic. Use `IN` for incoming (see One Time Pad IN).
`groupcount`	Number of groups per line. Default is `10`.

Related Topics

- Reports Overview
- One Time Pad IN
- One Time Pad OUT Simplified
- One Time Pad
- One Time Pads
- Print Templates

Report: One Time Pad OUT Simplified



2026-08-20 21:49:41.182678534

Period 1 / Day 1

Pad ID 89807 OUT 89807

89807	42618	03795	57423	37436
25364	12089	88862	94684	12680
61549	74604	23406	10332	56785
60335	84634	60140	99873	24200
50368	84289	36454	38455	46706
11818	51131	22083	33248	82791
63707	36253	53081	11561	71886
35258	68057	61875	38683	79492
89751	01609	99094	68107	03095
67592	78730	30303	12546	39705
50340	27161	65204	36714	11221
51204	98074	29490	19386	91014
54352	28506	75686	44369	85892
00147	94037	18257	71164	05947
08727	64790	96260	82514	44679
11933	98301	96301	96649	43862
36268	14842	25850	63766	17872
62748	34277	16082	31932	62270
88081	07472	53797	57723	70594
81718	48337	82768	62488	02142

Pad ID 65121 OUT 65121

65121	18788	69895	70392	34420
08596	35960	34276	29328	25010

Period 1 / Day 1

Pad ID 65121 OUT Cont... 65121

42541	34024	96165	48328	84869
61681	34163	20311	81208	32268
14772	80543	48822	35946	79581
74955	50220	86308	70273	49840
04044	01840	94096	30732	21015
08855	35547	03413	61821	73805
53437	43810	65872	59762	09152
40829	40830	12549	51336	46779
33551	07887	12459	79460	46141
48087	40872	33856	93556	65378
65229	85379	20035	26270	56503
67241	06263	36850	49644	71141
63147	24040	78821	74616	12802
75028	97183	78983	25346	25837
76875	90038	59174	55491	76976
47563	73106	63207	45080	47590
70057	77767	73718	66685	34485
22457	95675	18502	76893	66361

Official Use Only

OneTimePadSimple.jrxml

2026-08-20 21:49:41.182678534

Prints a simplified outgoing one-time pad with 5 groups per line. The wider spacing makes individual groups easier to read and reduces transcription errors during transmission.

Template Details

Field	Value
Template File	`OneTimePadSimple.jrxml`
Template Type	Jasper Reports
Data Source	ONE_TIME_PAD
Properties	`groupcount=5` `direction=OUT`

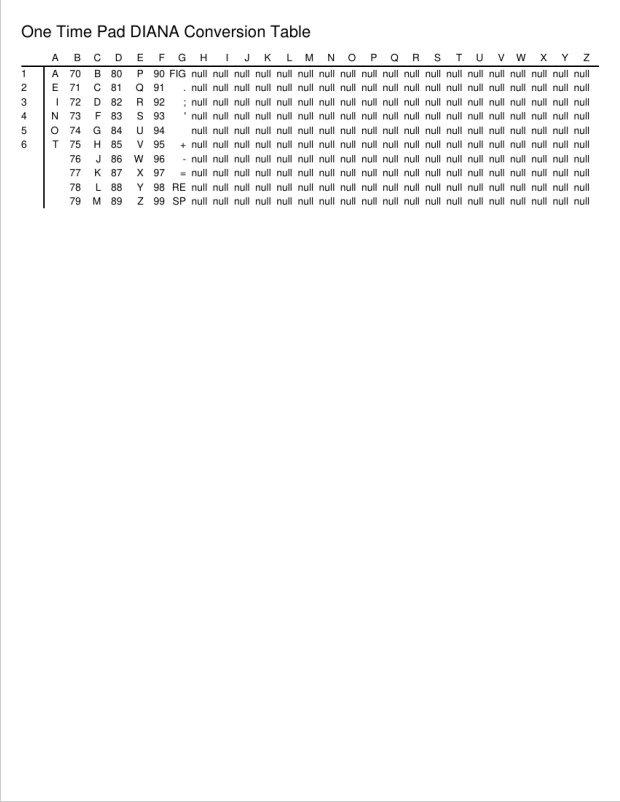
Properties

Property	Description
`groupcount`	Number of groups per line. Default is `5` for this template.
`direction`	Directional label on the pad header (`IN` or `OUT`).

Related Topics

- Reports Overview
- One Time Pad OUT
- One Time Pad IN Simplified
- One Time Pads
- Print Templates

Report: OTP DIANA Conversion Table


One Time Pad DIANA Conversion Table

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	70	B	80	P	90	FIG	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
2	E	71	C	81	Q	91	.	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
3	I	72	D	82	R	92	;	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
4	N	73	F	83	S	93	'	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
5	O	74	G	84	U	94		null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
6	T	75	H	85	V	95	+	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
		76	J	86	W	96	-	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
		77	K	87	X	97	=	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
		78	L	88	Y	98	RE	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
		79	M	89	Z	99	SP	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null

Prints the DIANA codex — the lookup table used to convert between letters and digits when working with DIANA-type one-time pads. This table is typically included alongside DIANA OTP pads so operators can perform letter-based encryption and decryption.

Template Details

Field	Value
Template File	`OTPDianaConversionTable.jrxml`
Template Type	Jasper Reports
Data Source	OTP_NUMBER_CODEX
Properties	*None*

Output

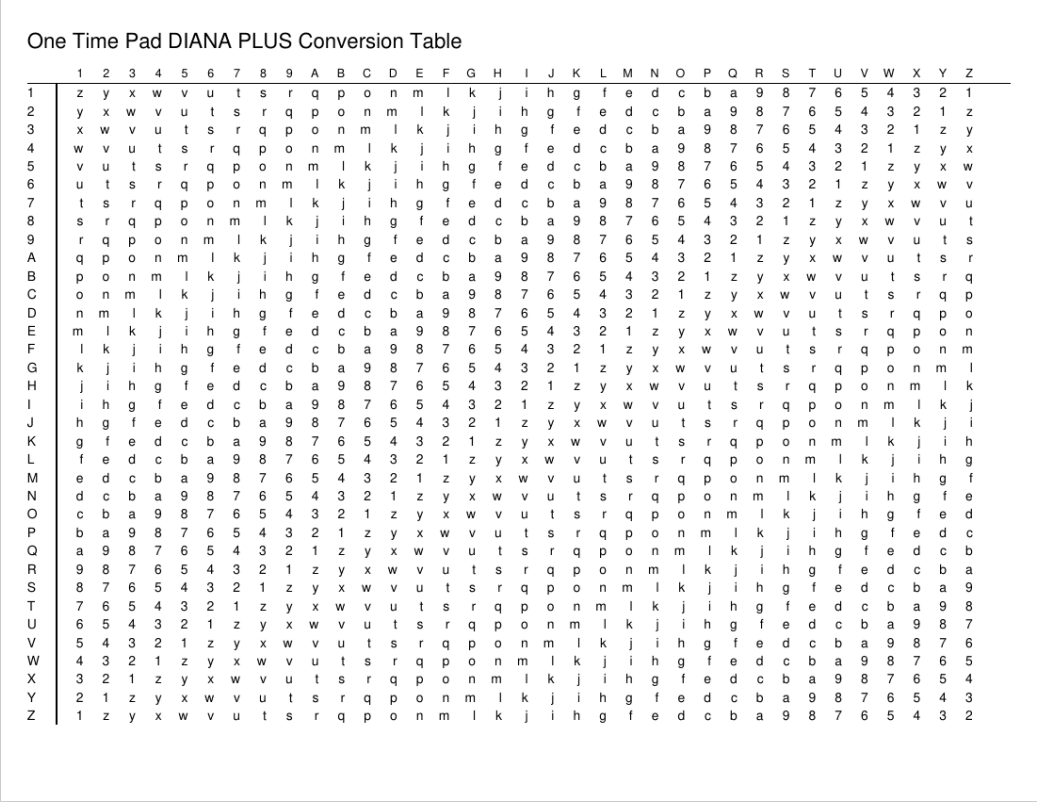
A formatted codex grid showing the DIANA letter-to-digit and digit-to-letter mapping. One page is sufficient; this report is typically paired with One Time Pad or One Time Pad IN in a Print Configuration.

Related Topics

- Reports Overview
- OTP Diana Plus Conversion Table

- OTP Numeric Conversion Table
- One Time Pads
- Print Templates

Report: OTP Diana Plus Conversion Table



One Time Pad DIANA PLUS Conversion Table

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1
2	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z
3	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y
4	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x
5	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w
6	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v
7	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u
8	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t
9	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s
A	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r
B	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q
C	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p
D	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o
E	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n
F	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m
G	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l
H	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k
I	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j
J	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i
K	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h
L	f	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g
M	e	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f
N	d	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e
O	c	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d
P	b	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c
Q	a	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b
R	9	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a
S	8	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9
T	7	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8
U	6	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7
V	5	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6
W	4	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5
X	3	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4
Y	2	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3
Z	1	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	9	8	7	6	5	4	3	2

Prints the DIANA+ reciprocal codex — the conversion table used with DIANA+ one-time pads. DIANA+ supports both letters and numbers, and its codex includes both the forward (encrypt) and reverse (decrypt) mappings in a single table.

Template Details

Field	Value
Template File	`OTPDianaPlusConversionTable.jrxml`
Template Type	Jasper Reports
Data Source	OTP_DIANA_PLUS_RECIPROCAL_CODEX
Properties	*None*

Output

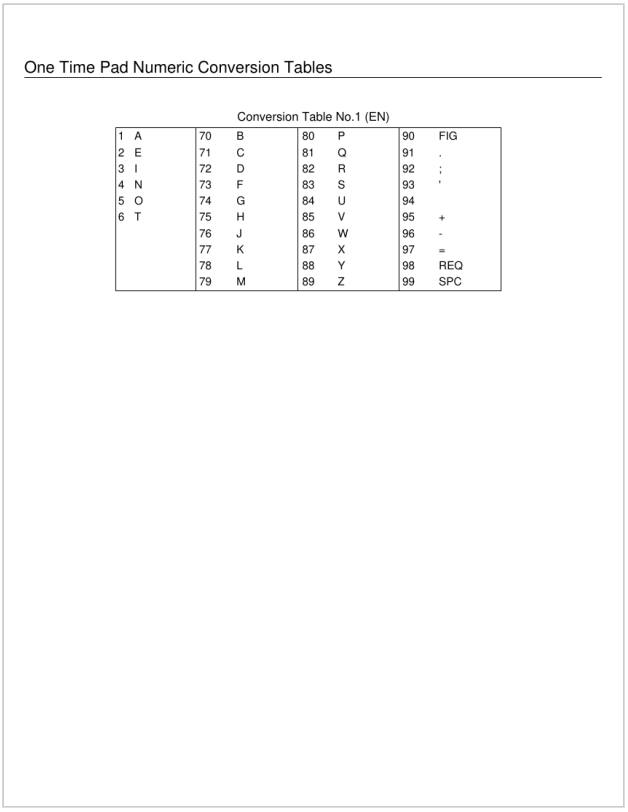
A formatted DIANA+ reciprocal codex grid. Include this in a Print Configuration alongside DIANA+ OTP pads so operators have the conversion reference they need.

Related Topics

- Reports Overview
- OTP DIANA Conversion Table

- OTP Numeric Conversion Table
- One Time Pads
- Print Templates

Report: OTP Numeric Conversion Table



One Time Pad Numeric Conversion Tables

Conversion Table No.1 (EN)

1	A	70	B	80	P	90	FIG
2	E	71	C	81	Q	91	.
3	I	72	D	82	R	92	;
4	N	73	F	83	S	93	'
5	O	74	G	84	U	94	
6	T	75	H	85	V	95	+
		76	J	86	W	96	-
		77	K	87	X	97	=
		78	L	88	Y	98	REQ
		79	M	89	Z	99	SPC

Prints the numeric OTP codex table. This reference shows the digit-group encoding scheme used with numeric one-time pads to convert plaintext numbers to ciphertext groups and back.

Template Details

Field	Value
Template File	`OTPNumericCodex.jrxml`
Template Type	Jasper Reports
Data Source	OTP_NUMBER_CODEX
Properties	*None*

Output

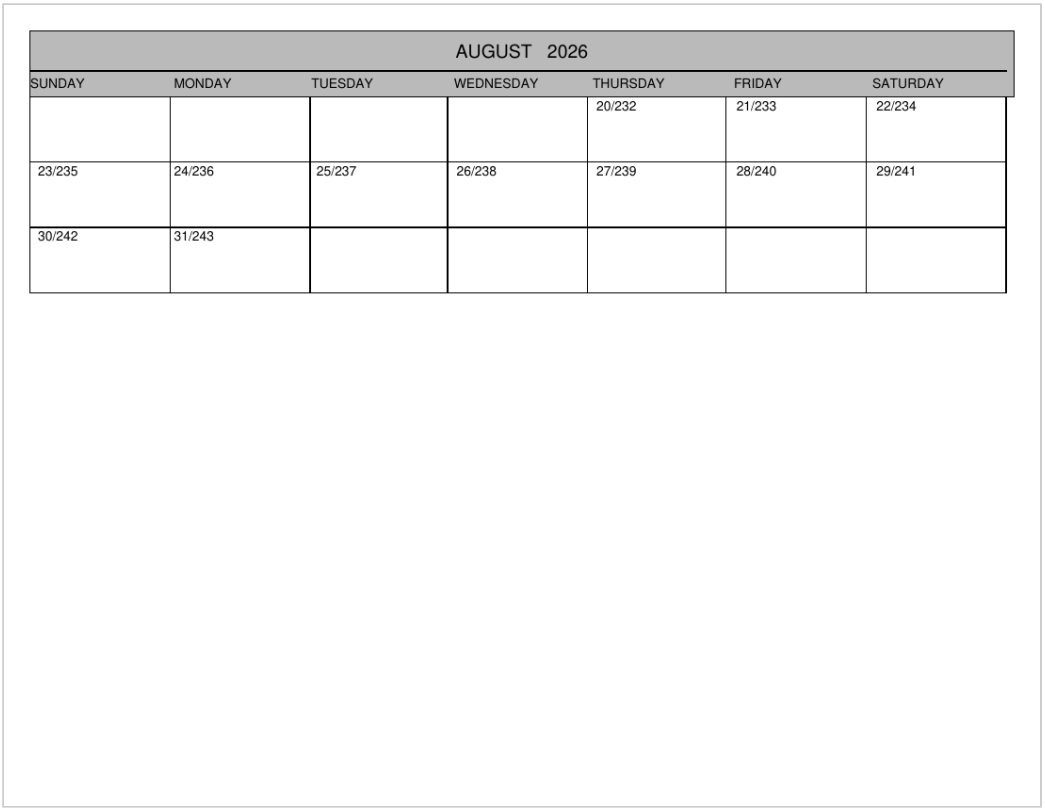
A single-page codex grid for numeric OTP operations. Include this alongside numeric OTP pads in a Print Configuration.

Related Topics

- Reports Overview
- OTP DIANA Conversion Table
- OTP Diana Plus Conversion Table

- One Time Pads
- Print Templates

Report: Month Calendar



AUGUST 2026

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
				20/232	21/233	22/234
23/235	24/236	25/237	26/238	27/239	28/240	29/241
30/242	31/243					

Prints a landscape monthly calendar covering 12 months from the current date. Useful as a planning reference to annotate with CEOI period and division dates, communications windows, or operational milestones.

Template Details

Field	Value
Template File	`MonthCalendarLandscape.jrxml`
Template Type	Jasper Reports
Data Source	CALENDAR
Properties	*None*

Output

Twelve landscape calendar pages, one per month, starting from the current month at print time. Days of the week begin on Sunday. Cells are sized to allow hand-written annotations.

> Copy `templates/jasper/MonthCalendarLandscape.jrxml` to change the week start day, add pre-printed period labels, or adjust the cell size for a specific paper format.

Related Topics

- Reports Overview
- Print Templates

Reports – Advanced: Data Source Reference

Every Jasper Reports template receives its data through a **data source**: a small JavaScript file that reads the open CEOI and returns rows. Each row is a set of named values, and each name becomes a report field the template refers to as `$F{fieldName}`.

This section documents every data source that ships with Scepter and the fields it produces. Use it when writing or modifying templates. To write a data source of your own, see Writing Custom Data Sources.

How Data Sources Work

When a Print Job runs, Scepter looks up the `dataSource` id configured on the template, runs that data source's script against the open CEOI, and hands the rows to Jasper as JSON. Jasper iterates the rows, exposing each name as a report field.

The scripts live in two places. The shipped ones are in `templates/datasources/` and are replaced when Scepter is updated. Your own are in `config/datasources/` and are never touched by an update. Both are listed in Report Data Sources, where you can read any of them.

Field Names

A field name is whatever the script called it — there is no getter-name convention to translate. What the script writes as `{ callSign: "ALPHA" }` the template reads as `$F{callSign}`. The names below are exactly what each data source produces.

Field Values Are Scalars

A field value is a single value: text, a number, or a true/false. It cannot be a list. Where a report needs a list — a group's members, an SOI's PACE plan, a pad's lines — the data arrives as **nested rows**, reached by a subreport or list component whose data source expression is:

```
((net.sf.jasperreports.json.data.JsonQLDataSource)$P{REPORT_DATA_SOURCE}).subDataSour
```

Nested rows are noted as such in the tables below. They are not fields, so a template cannot call methods on them: a report that needs to know how many there are wants a scalar count alongside, the way `UNITS` provides `groupCount`.

Designing a Template Against Real Data

Any data source can write its rows to a JSON file that Jasper Studio can open directly. Select it in Report Data Sources and click **Export Data...**. It is the same JSON the print path feeds Jasper, so a template designed against the file fills correctly the first time it runs inside Scepter. The exported file holds live CEOI data and is written unencrypted — look after it as you would a printed CEOI.

Available Data Sources

Data Source Id	Script	Used By

WORDLIST_ENCODE / WORDLIST_DECODE	`wordlist-encode.js`, `wordlist-decode.js`	Wordlist Encode, Wordlist Decode
UNITS	`units.js`	Unit Detail
UNIT_CALLSIGNS	`unit-callsigns.js`	Call Signs
UNIT_AUTHENTICATION_CODES_BY_PERIOD	`unit-authentication-codes-by-period.js`	Authentication Codes
UNIT_AUTHENTICATION_CODES_BY_UNIT	`unit-authentication-codes-by-unit.js`	No shipped report — the same rows grouped by unit, for your own template
UNIT_PASSWORDS	`unit-passwords.js`	Passwords
FREQUENCY_ENCODE	`frequency-encode.js`	Frequency Codes – Encrypt
UNIT_DAILY_CODES	`unit-daily-codes.js`	Daily Codes (Single), Daily Codes (Multi)
UNIT_CODES_CONDENSED	`unit-codes-condensed.js`	Daily Codes Condensed
GROUPS	`groups.js`	Group Roster
CEOI	`ceoi.js`	Cover Sheet, Basic Text Cover Sheet
LOCATION_ENCODE / LOCATION_DECODE	`location-encode.js`, `location-decode.js`	Location Codes Encrypt/Decrypt
DOCUMENT_ENCODE / DOCUMENT_DECODE	`document-encode.js`, `document-decode.js`	Document Codes Encrypt/Decrypt
FREQUENCY_DECODE	`frequency-decode.js`	Frequency Codes – Decrypt
CYPHERTABLES	`cipher-tables.js`	DRYADs

SARNEG	`sarnegs.js`	SARNEGs
CHALLENGE_RESPONSE	`challenge-response.js`	Challenge Response Words
ONE_TIME_PAD	`one-time-pad.js`	All One Time Pad reports
OTP_NUMBER_CODEX	`otp-number-codex.js`	OTP Numeric Conversion Table
OTP_DIANA_RECIPROCAL_CODEX	`otp-diana-codex.js`	OTP Diana Conversion Table
OTP_DIANA_PLUS_RECIPROCAL_CODEX	`otp-diana-plus-codex.js`	OTP Diana Plus Conversion Table
SOI	`soi.js`	SOI
CALENDAR	`calendar.js`	Month Calendar
FREQUENCIES, LOCATIONS, DOCUMENTS	`frequencies.js`, `locations.js`, `documents.js`	The plain entity list reports

Related Topics

- Writing Custom Data Sources
- Report Data Sources
- Reports Overview
- Print Templates

Data Source: WORDLIST_ENCODE / WORDLIST_DECODE

Both word list data sources produce a flat list of rows — one row per word per period and division. The difference is sort order: **WORDLIST_ENCODE** sorts by plaintext word (for looking up a word to find its code), while **WORDLIST_DECODE** sorts by code (for looking up a code to find its word).

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) to which this word/code pair belongs.
`division`	`int`	Division number within the period (1-based).
`word`	`String`	The plaintext word or phrase from the word list.
`code`	`String`	The generated code assigned to this word for this period/division.
`groupChar`	`String`	Single uppercase character used for alphabetical grouping in the report. For WORDLIST_ENCODE this is the first letter of `word`; for WORDLIST_DECODE it is the first letter of `code`.

Sorting

Data Source	Primary Sort	Secondary Sort
WORDLIST_ENCODE	Period	Plaintext word (A–Z)
WORDLIST_DECODE	Period	Code (A–Z)

Usage Notes

Use `$F{groupChar}` as a group expression in the report to produce alphabetical section breaks. There is one row per word per period/division combination, so a CEOI with 3 periods and 2 divisions per period will produce 6 rows for each word in the list.

Used By

- Wordlist Encode
- Wordlist Decode

Related Topics

- Data Source Reference
- Word List

Division-Column Code Grids

Nine data sources share one row shape: **one row per entity per period**, with the divisions laid out across the row as up to ten code columns. This is the shape behind every report that prints a code table with a column per day.

Data Source	Entity	Notes
`UNIT_CALLSIGNS`	Unit / Group	Call signs.
`UNIT_AUTHENTICATION_CODES_BY_PERIOD`	Unit / Group	Authentication codes, sorted period first.
`UNIT_AUTHENTICATION_CODES_BY_UNIT`	Unit / Group	The same rows sorted unit first, for a template that groups by unit. No shipped report uses it; it is there for your own.
`UNIT_PASSWORDS`	Unit / Group	Passwords.
`FREQUENCY_ENCODE`, `FREQUENCIES`	Frequency	Identical rows; the two exist so one template can group by period and the other not.
`LOCATION_ENCODE`, `LOCATIONS`	Location	As above.
`DOCUMENT_ENCODE`, `DOCUMENTS`	Document	As above.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this row.
`number`	`int`	Sequential row number within the period, useful for zebra striping or row numbering.
`name`	`String`	Display name of the entity (unit, group, frequency, location or document).
`groupName`	`String`	Name of the parent group context. Populated when codes are generated in the context of a group membership; blank otherwise.
`isGroup`	`boolean`	`true` if the entity is a Group, `false` otherwise.
`other`	`String`	

		Additional descriptive text; populated by some sources for supplementary information.
`detail`	`String`	Entity detail or description — a frequency's value and mode, a location's coordinates, a document's description. **Entity sources only.** The three unit sources do not emit it.
`value`	`String`	The entity's own value, alongside the codes that stand for it. **Entity sources only.**
`column1` – `column10`	`String`	Generated code for this entity in division 1 through division 10 of this period. Empty string if the entity has no code for that division, or if the CEOI has fewer divisions than the column index.
`column1Title` – `column10Title`	`String`	Header label for the corresponding code column, formatted from the CEOI period/division configuration (e.g., “Day 1”, “Shift 2”). Use these as column headers in the report band.

> `detail` and `value` are present on the frequency, location and document sources and absent from `UNIT_CALLSIGNS`, the two authentication sources and `UNIT_PASSWORDS`. A template written against one family and pointed at the other will show those two fields empty.

Column Layout

The number of meaningful columns equals the number of divisions in the selected period. Columns beyond the division count return empty string. A report template should use `$F{column1Title}` etc. to label the columns rather than hard-coding division names.

Sorting

Period, then entity name (case-insensitive) — except `UNIT_AUTHENTICATION_CODES_BY_UNIT`, which sorts period, name, containing group, then row number.

Used By

- Call Signs (`UNIT_CALLSIGNS`)
- Authentication Codes (`UNIT_AUTHENTICATION_CODES_BY_PERIOD`)
- Passwords (`UNIT_PASSWORDS`)
- Frequency Codes – Encrypt (`FREQUENCY_ENCODE`)
- Location Codes – Encrypt (`LOCATION_ENCODE`)
- Document Codes – Encrypt (`DOCUMENT_ENCODE`)

Related Topics

- Data Source Reference
- Code Lookup Data Sources -- the Decrypt side
- UNIT_DAILY_CODES Data Source

Data Source: UNIT_DAILY_CODES

Produces a flat list of rows — one row per actor (unit or group) per period per division. Each row carries the callsign, authentication code, and password assigned to that actor for that specific time window, making it suitable for daily COMSEC card layouts where each division occupies its own section or page.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based).
`division`	`int`	Division number within the period (1-based).
`name`	`String`	Display name of the actor (unit or group).
`groupName`	`String`	Parent group context name. When codes are generated in the context of a group membership, this identifies the group. Blank when codes are generated at the CEOI level.
`isGroup`	`boolean`	`true` if the actor is a Group; `false` if a Unit.
`number`	`int`	Sequential row number within the division, useful for row numbering or striping.
`callSign`	`String`	Callsign assigned to this actor for this period/division. Empty string if none generated.
`authCode`	`String`	Authentication code assigned to this actor for this period/division. Empty string if none generated.
`password`	`String`	Password assigned to this actor for this period/division. Empty string if none generated.
`other`	`String`	Reserved for additional code fields. Currently empty; available for custom factory extensions.

Row Multiplicity

The dataset contains one row for every combination of (actor × period × division). A CEOI with 10 units, 3 periods, and 2 divisions per period produces 60 rows. Group by `$F{period}` and `$F{division}` in the report to produce per-day sections.

The **Multi Divisions Per Page** template groups multiple divisions together; the **Single Division Per Page** template uses one page break per division. Both use the same dataset.

Used By

- Daily Codes (Single Division Per Page)
- Daily Codes (Multi Divisions Per Page)

Related Topics

- Data Source Reference
- UNIT_CODES_CONDENSED Data Source
- EntityCodesTableRow Data Source

Data Source: UNIT_CODES_CONDENSED

Produces one row per item for the condensed daily codes layout. Unlike `DailyCodesTableRow` (which has fixed callsign/auth/password columns), this dataset produces **multiple rows per actor per division** — one row for each line of text that will appear in the condensed output. Each row carries a line heading (the code type label) and the corresponding line data (the code value), enabling a compact single-column text layout.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based).
`division`	`int`	Division number within the period (1-based).
`name`	`String`	Display name of the actor (unit or group).
`groupName`	`String`	Parent group context name, or blank if codes are CEOI-level.
`isGroup`	`boolean`	`true` if the actor is a Group.
`detailGroupValue`	`String`	Concatenation of `name` + `groupName`. Used as a group expression to keep all lines for the same actor together in the report.
`lineHeading`	`String`	Label for this line of codes (e.g., *CS:* for callsign, *AC:* for auth code, *PW:* for password). Rendered in a bold or distinct style in the condensed layout.
`lineData`	`String`	The actual code text for this line, potentially wrapping across the line if the value is longer than `maxLineLength`. Multiple codes may appear space-separated on one line.

Row Multiplicity

Each actor generates multiple rows per period/division: typically one row per code type (callsign, auth code, password). Group by `$F{detailGroupValue}` to keep all lines for one actor together. The `maxLineLength` property controls where line data wraps.

Used By

- Daily Codes Condensed

Related Topics

- Data Source Reference

- UNIT_DAILY_CODES Data Source

Data Source: UNITS

Produces one row per item — one bean per unit. Each bean carries the unit's name, number, custom fields, notes, and two nested rows for the unit's group memberships and assigned assets. This is the richest per-unit dataset and is designed for a comprehensive unit detail sheet.

Fields

Field (`$F{}`)	Type	Description
`name`	`String`	Unit name.
`number`	`Integer`	Unit number (operator-assigned integer identifier; may be null).
`notes`	`String`	Free-text notes for this unit. Empty string if none.
`field_a`	`String`	Value of the unit's custom field A. The label for this field is controlled by the CEOI-level setting returned by `field_a_name`.
`field_b`	`String`	Value of the unit's custom field B.
`field_c`	`String`	Value of the unit's custom field C.
`field_a_name`	`String`	CEOI-level label for custom field A (configured in CEOI settings). Use as a column header.
`field_b_name`	`String`	CEOI-level label for custom field B.
`field_c_name`	`String`	CEOI-level label for custom field C.
`groupCount`	`Integer`	How many groups this unit belongs to. A scalar, so the template can hide the "Member Of" section when it is zero.
`groups`	nested rows	The groups this unit belongs to, each shaped like an asset row (`assetName` , `assetTypeString` , `assetIdentifier` , `assetPeriod` , `assetDivision`). Reached with `subDataSource("groups")` , not as a field.
`assets`	nested rows	The assets assigned to this unit, same shape as `groups` above. Reached with `subDataSource("assets")` , not as a field.

Sub-Dataset: Group

Each element of the `groups` sub-dataset is a `Group` entity. The most commonly used field is `$F{name}` (the group name). Other Group fields are accessible as standard bean properties.

Sub-Dataset: Asset

Each element of the `assets` sub-dataset is an `Asset` entity (which may be a DRYAD, OTP, SARNEG, ChallengeResponse, or SOI). The `$F{name}` field gives the asset's display name; use `$F{assetType}` to distinguish asset types.

Used By

- Unit Detail

Related Topics

- Data Source Reference
- Units

Data Source: GROUPS

Produces one row per item — one bean per group. Each bean carries the group name and a nested sub-dataset of the group's member units. This dataset drives the Group Roster report.

> The `selected` property (a comma-separated list of group UUIDs) can be set on the template to restrict the report to specific groups. If absent or empty, all groups are included.

Fields

Field (`$F{}`)	Type	Description
`name`	`String`	Group name. Rows are sorted alphabetically by this field.
`members`	nested rows	The units belonging to this group, sub-groups flattened in, each with a `name`. Reached with `subDataSource("members")`, not as a field.

Sub-Dataset: Unit Members

Each element of `membersDS` is a `Unit` entity. The fields most useful in a roster sub-report are:

Field (`$F{}`)	Description
`name`	Unit name.
`number`	Unit number (Integer, may be null).
`detail`	Unit detail/description field.

Used By

- Group Roster

Related Topics

- Data Source Reference
- Groups

Data Source: CEOI

The CEOI data source passes an **empty row dataset** (`JREmptyDataSource(1)`) to the template — exactly one empty iteration. All CEOI data is instead injected into the report's **parameters map** and accessed via `$P{}` expressions rather than `$F{}` field expressions. This is the appropriate data source for cover sheets and other reports that display document-level metadata rather than repeating per-entity data.

Available Parameters (`$P{}`)

The following parameters are populated from the open CEOI document. In the Jasper Studio parameter editor, set the parameter type to `Object` or the appropriate Java type listed below.

Parameter	Type	Description
`CEOI`	`CEOI`	The full CEOI entity object. Gives access to all document properties via method calls in expressions (e.g., `$P{CEOI}.getTitle()`).
`REPORT_CONNECTION`	`Connection`	Standard Jasper parameter; not used by Scepter but present in every report context.

Commonly Used CEOI Properties

Access these via `$P{CEOI}.getXxx()` in a text field expression:

Expression	Description
`$P{CEOI}.getTitle()`	CEOI document title.
`$P{CEOI}.getDistribution()`	Distribution line (e.g., *UNCLASSIFIED*).
`$P{CEOI}.getSerial()`	Document serial number.
`$P{CEOI}.getStartDate()`	CEOI effective start date.
`$P{CEOI}.getEndDate()`	CEOI effective end date.
`$P{CEOI}.getPeriodCount()`	Number of periods in the CEOI.
`$P{CEOI}.getDivisionUnitsString()`	The label used for divisions (e.g., “Day”, “Shift”).

FreeMarker Equivalent

In FreeMarker templates (`.ftlh`), the CEOI object is available as the variable `CEOI` and accessed directly: `${'$'}{CEOI.title}`, `${'$'}{CEOI.distribution}`, etc.

Used By

- Cover Sheet (pdf)

- Basic Text Cover Sheet

Related Topics

- Data Source Reference
- Cover Settings

Data Source: LOCATION_DECODE / DOCUMENT_DECODE / FREQUENCY_DECODE

These three data sources produce a flat code-to-entity lookup — one row per entity per period per division, sorted by code. They are what a receiving station reads: it has a code off the air and needs the location, document or frequency it stands for.

> The matching *Encrypt* data sources are a different row shape, not a different sort order. They lay the divisions out as columns rather than as rows, and carry no `code` field at all. See Division-Column Code Grids.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this code assignment.
`division`	`int`	Division number within the period (1-based).
`code`	`String`	The generated code assigned to this entity for this period/division.
`name`	`String`	Display name of the entity (location name, document name, or frequency name).
`detail`	`String`	Additional entity detail. Content depends on the entity type: • **Location:** coordinate string (Lat/Long, MGRS, UTM, etc.) • **Document:** document description or filename • **Frequency:** frequency value and mode (e.g., “146.520 FM”)

Sorting

Period → Division → Code (A–Z). Sorting by code is the whole point of these three: it makes the printed sheet something you can scan down to find a received code.

Usage Notes

Group by `$F{period}` and optionally `$F{division}` to produce per-period sections in the report. The `detail` field allows a two-column reference table for locations (code ↔ coordinates) or frequencies (code ↔ frequency+mode) without needing a sub-dataset.

Used By

- Location Codes – Decrypt
- Document Codes – Decrypt
- Frequency Codes – Decrypt

Related Topics

- Data Source Reference
- Division-Column Code Grids -- the Encrypt side

Data Source: CALENDAR

One row per month, each carrying its weeks as nested rows. It is the only shipped data source that does not read the CEOI at all — every value comes from the properties below, so the same report prints the same calendar for any CEOI.

Fields

Field (`$F{}`)	Type	Description
`month`	`int`	Month number, 1–12.
`monthName`	`String`	Month name in capitals, e.g. “JANUARY”.
`year`	`int`	Four-digit year.
`day1Name` – `day7Name`	`String`	Weekday names for the seven columns, in the order set by the `weekstart` property. Use these as the column headers.

Nested Rows

`weeks` is a nested row set, not a field — one row per week line of the month. Reach it from a subreport or list component whose data source expression is:

```
((net.sf.jasperreports.json.data.JsonQLDataSource)$P{REPORT_DATA_SOURCE}).subDataSour
```

Nested Field	Description
`day1DisplayString` – `day7DisplayString`	What to print in that column: day of month and Julian day of year, or empty for a cell before the first or after the last day of the month.

Properties

Set these on the print template or the print job, the same as any other data source property.

Property	Default	Description
`startdate`	today	First month to print, as `yyyy-mm-dd`.
`months`	`12`	How many months to print, one row — and so one page — each.
`weekstart`	`SUNDAY`	Which weekday is the first column.

Used By

- Month Calendar

Related Topics

- Data Source Reference
- Writing Custom Data Sources

Data Source: CYPHERTABLES

Produces one row per item — one bean per DRYAD. Each bean exposes the table's metadata fields plus **25 fixed row accessors** (`row0` through `row24`), each returning a `DRYADRow` object that represents one row of the grid. This layout maps directly to the fixed 10×25 DRYAD grid format.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this DRYAD.
`division`	`int`	Division number within the period (1-based).
`number`	`int`	Sequential number of this DRYAD within the period/ division.
`name`	`String`	Display name of the DRYAD table.
`divisionWord`	`String`	The label used for divisions in this CEOI (e.g., "Day", "Shift"), from CEOI settings.
`sHA1`	`String`	SHA-1 hash of the DRYAD table data. Use for integrity verification or printing alongside the table.
`sHA256`	`String`	SHA-256 hash of the DRYAD table data.
`sHA512`	`String`	SHA-512 hash of the DRYAD table data.
`row0` – `row24`	`DRYADRow`	One row of the cipher grid. Each `DRYADRow` object exposes the cell values for that row. See the DRYADRow field table below.

Sub-Object: `DRYADRow` Fields

Access `DRYADRow` fields via expression syntax, e.g.: `$F{row0}.getCell(0)` or by accessing the row object in a sub-expression.

Method	Description
`getRowLabel()`	The label for this grid row (typically a digit 0–9, cycling across the 25 rows).
`getCell(int col)`	Returns the letter value at column index `col` (0-based, 0–9 for the 10 columns of the DRYAD grid).

Grid Dimensions

A standard DRYAD table has 10 columns (representing digits 0–9) and 25 rows. The 25 rows are addressed as `$F{row0}` through `$F{row24}`. This fixed structure means every DRYAD report template must reference all 25 row fields explicitly rather than using a repeating band.

Used By

- DRYADs

Related Topics

- Data Source Reference
- DRYADs
- DRYAD Viewer

Data Source: SARNEG

Produces one row per item — one per SARNEG table. Each bean exposes the SARNEG metadata and its **10 key letters** as individual fields (`letter0` through `letter9`). The letters are the individual characters of the SARNEG keyword, accessed one-by-one to populate the grid column headers.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this SARNEG.
`division`	`int`	Division number within the period (1-based).
`number`	`int`	Sequential number of this SARNEG within the period/division.
`name`	`String`	Display name of the SARNEG table.
`word`	`String`	The full 10-character SARNEG keyword. This is the same string that `letter0`–`letter9` decompose character by character.
`divisionWord`	`String`	The label used for divisions in this CEOI (e.g., “Day”, “Shift”).
`letter0`	`String`	First character of the SARNEG keyword. Used as the column-0 header of the grid.
`letter1`	`String`	Second character of the keyword (column 1 header).
`letter2`	`String`	Third character (column 2 header).
`letter3`	`String`	Fourth character (column 3 header).
`letter4`	`String`	Fifth character (column 4 header).
`letter5`	`String`	Sixth character (column 5 header).
`letter6`	`String`	Seventh character (column 6 header).
`letter7`	`String`	Eighth character (column 7 header).
`letter8`	`String`	Ninth character (column 8 header).
`letter9`	`String`	Tenth (last) character of the keyword (column 9 header).

Usage Notes

A standard SARNEG grid has 10 columns, one per letter of the keyword, and 10 rows (digits 0–9). Place `$F{letter0}` through `$F{letter9}` as column headers. The grid cell values are derived by the report template from the column letters and row digits using the standard SARNEG encoding rule.

Used By

- SARNEGs

Related Topics

- Data Source Reference
- SARNEGs
- SARNEG Viewer

Data Source: CHALLENGE_RESPONSE

Produces one row per item — one per challenge/response pair. Each bean exposes the three words of the pair (challenge, response, duress) plus period and division identifiers, making it straightforward to print a reference card grouped by time window.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this pair.
`division`	`int`	Division number within the period (1-based).
`number`	`int`	Sequential number of this pair within the period/division.
`divisionWord`	`String`	The label used for divisions in this CEOI (e.g., “Day”, “Shift”).
`id`	`String`	Display name of this challenge/response pair (as entered in the CEOI).
`challenge`	`String`	The challenge word. An operator issues this word; the other party must reply with the correct response word.
`response`	`String`	The correct response word. Giving this word proves identity to the challenger.
`duressWord`	`String`	The duress word. Using this instead of the response word signals to friendly forces that the responder is under coercion.

Usage Notes

Group by `$F{period}` and `$F{division}` to produce per-time-window sections. The three word fields can be placed side by side for a compact card or stacked with labels for a less ambiguous layout.

Used By

- Challenge Response Words

Related Topics

- Data Source Reference
- Challenge/Response

Data Source: ONE_TIME_PAD

Produces one row per item — one per one-time pad. Each bean exposes the pad's metadata and a nested sub-dataset of rows that carry the actual pad data as groups of 5 characters. The `groupcount` property controls how many groups appear per line; the `direction` property labels the pad as IN or OUT for directional variants.

Fields

Field (`$F{}`)	Type	Description
`period`	`int`	CEOI period number (1-based) for this pad.
`division`	`int`	Division number within the period (1-based).
`number`	`int`	Sequential pad number within the period/division.
`divisionWord`	`String`	The label used for divisions in this CEOI (e.g., "Day", "Shift").
`divisionString`	`String`	Same as `divisionWord` ; provided for compatibility with both field names used across templates.
`type`	`String`	OTP type string: *Numbers*, *DIANA*, or *DIANA+*.
`direction`	`String`	Direction label for directional pads: `"IN"` , `"OUT"` , or `null` for non-directional pads. Used to print a direction banner on the pad sheet.
`seriesName`	`String`	The pad's series/pad identifier string (from the OTP entity's pad ID field).
`padId`	`String`	The pad's key/serial identifier (from the OTP entity's key ID field). Used on the pad header to identify the specific pad sheet.
`size`	`int`	Total number of characters in this pad. Equal to the number of groups × 5.
`sHA1`	`String`	SHA-1 hash of the pad data. Print alongside the pad for integrity verification.
`sHA256`	`String`	SHA-256 hash of the pad data.
`sHA512A`	`String`	First 64 characters of the SHA-512 hash of the pad data (SHA-512 is 128 hex characters; split into two fields for layout reasons).
`sHA512B`	`String`	Last 64 characters of the SHA-512 hash.
`rows`	nested rows	

		One line of the pad sheet each. Reached with `subDataSource("rows")`, not as a field. See the nested row fields below.

Sub-Dataset: `OTPGroupsRow`

Each `OTPGroupsRow` represents one line of the printed pad, containing up to `groupcount` groups of 5 characters each. The number of groups per row is controlled by the `groupcount` template property (default: 10; simplified variants use 5).

Field (`$F{}`)	Type	Description
`group0`	`String`	First 5-character group on this line. Empty string if this line has fewer than 1 group.
`group1`	`String`	Second group. Empty string if fewer than 2 groups on this line.
`group2` – `group9`	`String`	Groups 3 through 10 on this line. Empty string if the line is shorter than that position. Standard pads use groups 0–9 (10 per line); simplified pads use 0–4 (5 per line).

> The last `OTPGroupsRow` in the sub-dataset may have fewer than `groupcount` groups if the pad size is not evenly divisible by the group count. Fields beyond the actual group count return empty string, so templates can safely reference all 10 group fields without conditional logic.

Used By

- One Time Pad
- One Time Pad IN
- One Time Pad OUT
- One Time Pad IN Simplified
- One Time Pad OUT Simplified

Related Topics

- Data Source Reference
- One Time Pads
- OTP Codex Data Source

Data Source: OTP_NUMBER_CODEX / OTP_DIANA_RECIPROCAL_CODEX / OTP_DIANA_PLUS_RECIPROCAL_CODEX

These three data sources all produce one row per item — one row per row of the codex table. Each bean carries an optional row label (`group`) and **36 column values** (`column0` through `column35`) that fill the cells of the conversion table grid. The number of columns actually used depends on the codex type.

The codex tables are reference grids that operators use alongside OTP pads to convert between letters and digits:

- **OTP_NUMBER_CODEX**: standard numeric codex used with Numbers and DIANA pads.
- **OTP_DIANA_RECIPROCAL_CODEX**: DIANA reciprocal codex (26-character alphabet).
- **OTP_DIANA_PLUS_RECIPROCAL_CODEX**: DIANA+ reciprocal codex (36-character alphabet: 26 letters + 10 digits).

Fields

Field (`$F{}`)	Type	Description
`group`	`String`	Row label or group identifier. For standard codex rows this is the digit or letter heading for the row (e.g., “0” for the row of digits beginning with 0). Empty string for rows with no label.
`column0`	`String`	Value in the first column (index 0) of this codex row.
`column1`	`String`	Value in column index 1.
`column2` – `column25`	`String`	Values for columns 2 through 25. Used by the 26-column DIANA codex.
`column26` – `column35`	`String`	Values for columns 26 through 35. Used only by the 36-column DIANA+ codex (which covers both letters and digits).

Column Count by Data Source

Data Source	Active Columns	Description
OTP_NUMBER_CODEX	0–9 (10 columns)	Numeric codex; one column per digit 0–9.
OTP_DIANA_RECIPROCAL_CODEX	0–25 (26 columns)	DIANA letter codex; one column per letter A–Z.

OTP_DIANA_PLUS_RECIPROCAL_CODEX	0–35 (36 columns)	DIANA+ codex; 26 letters + 10 digits.

Columns beyond the active range always return empty string, so templates that bind all 36 columns work safely with any codex type.

Used By

- OTP Numeric Conversion Table (OTP_NUMBER_CODEX)
- OTP DIANA Conversion Table (OTP_NUMBER_CODEX)
- OTP Diana Plus Conversion Table (OTP_DIANA_PLUS_RECIPROCAL_CODEX)

Related Topics

- Data Source Reference
- ONE_TIME_PAD Data Source
- One Time Pads

Data Source: SOI

Produces one row per item — one per SOI defined in the CEOI. Each bean exposes the SOI's metadata and three nested list accessors that drive sub-reports or table bands for the PACE plan, unit assignments, and communications schedule.

Fields

Field (`$F{}`)	Type	Description
`id`	`String`	UUID of the SOI entity (string form). Not typically printed; useful for conditional logic.
`title`	`String`	SOI title as entered in the SOI editor.
`objective`	`String`	Operational objective text for this SOI.
`location`	`String`	Name of the location associated with this SOI, or a "None" placeholder if no location is set.
`locationDetail`	`String`	Formatted string of all coordinate values for the associated location (Lat/Long, MGRS, UTM, Maidenhead, what3words), tab-separated, wrapping at 3 values per line. Empty if no location is set.
`duressCode`	`String`	Duress code assigned to this SOI.
`controlUnit`	`String`	Name of the net control unit for this SOI, or empty string if none assigned.
`controlUnitCallsign`	`String`	Callsign of the net control unit.
`recoveryUnit`	`String`	Name of the recovery unit, or empty string if none assigned.
`recoveryUnitCallsign`	`String`	Callsign of the recovery unit.
`sarneg`	`Sarneg`	The SARNEG entity associated with this SOI (if any). Access as

		`$F{sarneg}.getName()` etc. May be null; check before use.
`pacePlan`	`List<SOIPaceRow>`	List of exactly four PACE rows (PRIMARY, ALTERNATE, CONTINGENCY, EMERGENCY). Use as a sub-dataset. See `SOIPaceRow` below.
`unitAssignments`	`List<SOIUnitAssignmentRow>`	List of all unit assignments for this SOI. Use as a sub-dataset. See `SOIUnitAssignmentRow` below.
`commSchedule`	`List<SOICommScheduleRow>`	List of all scheduled communication windows across all days of this SOI. Use as a sub-dataset. See `SOICommScheduleRow` below.

Sub-Dataset: `SOIPaceRow` (from `pacePlan`)

Always exactly 4 rows in PRIMARY → ALTERNATE → CONTINGENCY → EMERGENCY order.

Field (`$F{}`)	Type	Description
`type`	`String`	Localized PACE type label: “Primary”, “Alternate”, “Contingency”, or “Emergency”.
`proword`	`String`	Proword associated with this PACE entry.
`signal`	`String`	Signal or authentication note for this PACE entry. Empty string if not set.
`txFreq`	`String`	Transmit frequency formatted string: frequency value, name, and mode (e.g., “146.520 Net1 FM”). Empty string if no frequency assigned (Emergency typically has none).
`rxFreq`	`String`	Receive frequency formatted string. Empty string if same as transmit or not assigned.

Sub-Dataset: `SOIUnitAssignmentRow` (from `unitAssignments`)

Field (`$F{}`)	Type	Description
`unitName`	`String`	Name of the assigned unit. Returns “N/A” if null or blank.

`callSign`	`String`	Callsign of the assigned unit in this SOI context. Returns “N/A” if null or blank.
`role`	`String`	Role of this unit in the SOI (e.g., “Net Member”, “Net Control”). Returns “N/A” if null or blank.

Sub-Dataset: `SOICommScheduleRow` (from `commSchedule`)

One row per communication window across all days of the SOI duration. Days are 1-based.

Field (`$F{}`)	Type	Description
`day`	`int`	Day number within the SOI (1-based).
`window`	`String`	Formatted time window string for this comm window (start time, duration, etc.).
`primary`	`String`	Primary frequency for this window: formatted as “frequency mode”, or “SOI Primary” / “SOI Alternate” if referencing the SOI’s own PACE entries.
`alternate`	`String`	Alternate frequency for this window, same format as `primary`. Empty string if none set.

Used By

- SOI

Related Topics

- Data Source Reference
- Signals Operating Instructions
- SOI Editor

FreeMarker Entity Reference: CEOI (Root Object)

FreeMarker templates receive the entire CEOI object as a top-level variable named `CEOI`. All public getter methods are accessible as properties using JavaBeans naming: `getTitle()` becomes `${CEOI.title}`, `isOtpPrintCodexes()` becomes `${CEOI.otpPrintCodexes?string}`. Methods that take parameters must be called explicitly: `${CEOI.getUnitById(id)}`.

Configuration Properties

Property	Java Type	Description
`title`	`String`	CEOI title.
`organization`	`String`	Organization name.
`revision`	`Integer`	CEOI revision number.
`periods`	`Integer`	Total number of periods.
`divisions`	`Integer`	Number of divisions per period.
`divisionUnits`	`DivisionUnits`	Division time unit enum: `DAY`, `WEEK`, `MONTH`, `QUARTER`, `YEAR`. Compare as string: `<#if CEOI.divisionUnits == "DAY">`
`divisionUnitsString`	`String`	Division unit label as a human-readable string (e.g., “Day”).
`distribution`	`CeoiDistribution`	Distribution level enum: `TRAINING`, `OFFICIAL_USE_ONLY`, `CONFIDENTIAL`, `CLASSIFIED`, `SECRET`, `TOP_SECRET`, `OTHER`.
`distributionString`	`String`	Distribution level as a display string.
`distributionOther`	`String`	Custom distribution text when `distribution == "OTHER"`.
`notes`	`String`	General CEOI notes.
`unitFieldA`	`String`	Label for custom unit field A (e.g., “Rank”).
`unitFieldB`	`String`	Label for custom unit field B.
`unitFieldC`	`String`	Label for custom unit field C.
`unitSeparator`	`String`	Separator character used between parts of a unit identifier.
`startDate`	`String`	CEOI start date as a formatted string.

`endDate`	`String`	CEOI end date as a formatted string.
`serial`	`String`	CEOI serial number string.
`periodCount`	`int`	Number of periods (same as `periods`; convenience accessor).

Cover Image

Property	Java Type	Description
`coverImage`	`String`	Cover image file name. May be null if none set.
`coverImageResourceId`	`UUID`	Resource identifier for the cover image.
`coverImageLocalFile`	`File`	Runtime file reference for the cover image.

Data Collections

Each collection is a `List` iterable with `<#list>`.

Property	Element Type	Description
`units`	`List<Unit>`	All units in the CEOI. See Unit.
`groups`	`List<Group>`	All groups. See Group.
`documents`	`List<Document>`	All document entities. See Document.
`frequencies`	`List<Frequency>`	All frequency entities. See Frequency.
`locations`	`List<Location>`	All location entities. See Location.
`words`	`List<Word>`	All word-list entries. See Word.
`sarnegs`	`List<Sarneg>`	All SARNEGs. See Sarneg.
`cipherTables`	`List<DRYAD>`	All DRYADs. See DRYAD.
`SOIs`	`List<SOI>`	All Signal Operating Instructions. See SOI.
`challengeResponse`	`List<ChallengeResponse>`	All challenge/response pairs. See ChallengeResponse.
`oneTimePads`	`List<OneTimePad>`	All one-time pads. See OneTimePad.
`codexList`	`List<Codex>`	All codex lookup tables. See Codex.

Lookup Methods

These return a single entity by UUID. Call them as methods since they take a parameter:

Call	Returns
`CEOI.getUnitById(id)`	`Unit` (or null)
`CEOI.getGroupById(id)`	`Group` (or null)
`CEOI.getLocationById(id)`	`Location` (or null)
`CEOI.getFrequencyById(id)`	`Frequency` (or null)
`CEOI.getDocumentById(id)`	`Document` (or null)
`CEOI.getWordById(id)`	`Word` (or null)
`CEOI.getDRYADById(id)`	`DRYAD` (or null)
`CEOI.getOTPById(id)`	`OneTimePad` (or null)
`CEOI.getSARNEGById(id)`	`Sarneg` (or null)
`CEOI.getSOIById(id)`	`SOI` (or null)
`CEOI.getChallengeResponseById(id)`	`ChallengeResponse` (or null)

Generation Settings

These properties expose the CEOI-level defaults that control code generation. Useful for informational cover sheets.

Property	Type	Description
`ciphertableMode`	`GenerationMode`	DRYAD generation scope: `CEOI`, `PERIOD`, or `DIVISION`.
`ciphertableCount`	`Integer`	Number of DRYADs to generate.
`wordlistMode`	`GenerationMode`	Word list code generation scope.
`wordListCodeType`	`WordlistCodeType`	`GENERATED` or `SEQUENCE`.
`wordListCodesCount`	`int`	Codes per word list entry.
`locationMode`	`GenerationMode`	Location code generation scope.
`locationAddToWords`	`Boolean`	Include locations in word list.
`locationsCodesCount`	`int`	Codes per location.
`frequencyMode`	`GenerationMode`	Frequency code generation scope.
`freqlistAddToWords`	`Boolean`	Include frequencies in word list.

`frequenciesCodesCount`	`int`	Codes per frequency.
`documentMode`	`GenerationMode`	Document code generation scope.
`documentAddToWords`	`Boolean`	Include documents in word list.
`documentsCodesCount`	`int`	Codes per document.
`sarnegMode`	`GenerationMode`	SARNEG generation scope.
`sarnegCount`	`Integer`	Number of SARNEGs to generate.
`challengeResponseMode`	`GenerationMode`	Challenge/response generation scope.
`challengeResponseCount`	`int`	Challenge/response pairs per division.
`oneTimePadMode`	`GenerationMode`	OTP generation scope.
`oneTimePadType`	`OTPType`	OTP type: `NUMBERS`, `DIANA`, or `DIANA_PLUS`.
`oneTimePadCount`	`Integer`	Number of OTPs to generate.
`oneTimePadSize`	`Integer`	OTP size in characters.
`otpPrintCodexes`	`boolean`	Whether to print conversion codexes alongside OTPs.

FreeMarker Usage Example

```
${CEOI.title} — Rev. ${CEOI.revision}
Distribution: ${CEOI.distributionString}
Periods: ${CEOI.periods} × ${CEOI.divisionUnitsString}s

<#list CEOI.units as unit>
  ${unit.name} (${unit.number})
</#list>
```

Related Topics

- Data Source Reference
- Unit Entity
- Group Entity
- DRYAD Entity
- SOI Entity
- OneTimePad & Codex Entities
- Basic Text Cover Sheet
- Cover Sheet

FreeMarker Entity Reference: Unit

Obtained from `CEOI.units` — a `List<Unit>`. Each `Unit` represents one operator or station. Units belong to one or more groups via `GroupMembership` objects, and each membership carries that unit's callsigns, authentication codes, and passwords for that group.

Unit Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Unit name.
`number`	`Integer`	Unit number.
`field_a`	`String`	Value for custom field A (label defined by `CEOI.unitFieldA`).
`field_b`	`String`	Value for custom field B.
`field_c`	`String`	Value for custom field C.
`notes`	`String`	Freeform notes for this unit.
`generate`	`boolean`	Whether codes are generated for this unit.
`groups`	`List<GroupMembership>`	All group memberships for this unit. See GroupMembership below.
`assetList`	`AssignedAssetList`	Assets assigned to this unit (DRYADs, OTPs, SARNEGs, etc.).
`assetCount`	`int`	Number of assigned assets.
`holderTypeName`	`String`	Always "Unit".
`actorIsGroup`	`boolean`	Always `false`.

Unit Methods with Parameters

These require explicit method-call syntax in FreeMarker:

Call	Returns	Description
`unit.getCallsigns(groupId)`	`List<GeneratedCode>`	Callsigns for a specific group (pass `null` for all).
`unit.getAuthCodes(groupId)`	`List<GeneratedCode>`	

		Authentication codes for a specific group.
`unit.getPasswords(groupId)`	`List<GeneratedCode>`	Passwords for a specific group.

Sub-Object: GroupMembership (from `unit.groups`)

Each entry in `unit.groups` represents membership in one group and holds the generated codes for that specific unit-in-group context.

Property	Java Type	Description
`groupId`	`UUID`	ID of the parent group. Use `CEOI.getGroupById(gm.groupId)` to resolve the full Group object.
`callsigns`	`List<GeneratedCode>`	Callsigns for this unit within this group. See GeneratedCode.
`authCodes`	`List<GeneratedCode>`	Authentication codes for this unit within this group.
`passwords`	`List<GeneratedCode>`	Passwords for this unit within this group.
`callsignCount`	`Integer`	Override callsign count (null = use CEOI default).
`authcodeCount`	`Integer`	Override auth code count (null = use CEOI default).
`passwordCount`	`Integer`	Override password count (null = use CEOI default).

Sub-Object: GeneratedCode

A single generated code value with its period/division context. Accessed from `groupMembership.callsigns`, `groupMembership.authCodes`, `groupMembership.passwords`, and similar properties on other entities.

Property	Java Type	Description
`code`	`String`	The generated code value.
`period`	`Integer`	Period this code is valid for (1-based).
`division`	`Integer`	Division within the period (1-based).
`number`	`int`	Sequential number of this code within its slot.
`override`	`boolean`	`true` if this code was manually overridden by the user.

`groupId`	`UUID`	Group this code belongs to.

FreeMarker Usage Example

```
<#list CEOI.units as unit>
${unit.number} ${unit.name}
  <#list unit.groups as gm>
    <#assign grp = CEOI.getGroupById(gm.groupId)>
    Group: ${grp.name}
    <#list gm.callsigns as cs>
      Period ${cs.period}, Div ${cs.division}: ${cs.code}
    </#list>
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- Group Entity
- Data Source Reference
- UNIT_CALLSIGNS Data Source (Jasper)
- UNITS Data Source (Jasper)

FreeMarker Entity Reference: Group

Obtained from `CEOI.groups` — a `List<Group>`. A `Group` is a named collection of units (and possibly sub-groups). It controls code generation settings for its members via a `GroupMemberCodeGenerationOptions` sub-object.

Group Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Group name.
`notes`	`String`	Freeform notes.
`units`	`List<UUID>`	UUIDs of direct member units. Resolve each with `CEOI.getUnitById(id)`.
`groups`	`List<UUID>`	UUIDs of child sub-groups. Resolve with `CEOI.getGroupById(id)`.
`memberOf`	`List<GroupMembership>`	Groups this group belongs to (parent memberships). See GroupMembership.
`memberCount`	`int`	Total count of direct unit and sub-group members.
`generationOptions`	`GroupMemberCodeGenerationOptions`	Code generation settings for this group's members. See below.
`assetList`	`AssignedAssetList`	Assets assigned to this group.
`assetCount`	`int`	Number of assigned assets.
`holderTypeName`	`String`	Always "Group".
`actorIsGroup`	`boolean`	Always `true`.

Sub-Object: GroupMemberCodeGenerationOptions (from `group.generationOptions`)

Controls how codes are generated for units that belong to this group.

Property	Java Type	Description

`generate`	`boolean`	Whether to generate codes for members of this group.
`generateForMembers`	`boolean`	Whether to generate individual codes for each member unit.
`globallyUnique`	`boolean`	Whether generated codes must be unique across the entire CEOI.
`unitCallsignPattern`	`String`	Pattern string used to generate callsigns for member units.
`unitCallsignCount`	`Integer`	Number of callsigns per unit (null = use CEOI default).
`unitPasswordsCount`	`Integer`	Number of passwords per unit.
`unitPasswordLength`	`Integer`	Character length of each password.
`unitPasswordChars`	`String`	Character set used when generating passwords.
`unitPasswordExcludeChars`	`String`	Characters explicitly excluded from password generation.
`unitPasswordNames`	`String`	Newline-separated list of names for each password slot (e.g., “Primary\nBackup”).
`unitAuthcodePattern`	`String`	Pattern for generating authentication codes.
`unitAuthcodeCount`	`Integer`	Number of authentication codes per unit.

FreeMarker Usage Example

```
<#list CEOI.groups as grp>
${grp.name} (${grp.memberCount} members)
  <#list grp.units as uid>
    <#assign u = CEOI.getUnitById(uid)>
    - ${u.name}
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- Unit Entity
- GROUPS Data Source (Jasper)

FreeMarker Entity Reference: Location

Obtained from `CEOI.locations` — a `List<Location>`. Each `Location` stores a named geographic point with multiple coordinate representations and a list of generated codes.

Location Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Location name.
`alternateName`	`String`	Alternate or short name.
`latitude`	`String`	Decimal latitude string.
`longitude`	`String`	Decimal longitude string.
`lat_lon`	`String`	Formatted “latitude longitude” combined string.
`mgrs`	`String`	MGRS grid coordinate.
`utm`	`String`	UTM coordinate string.
`maidenhead`	`String`	Maidenhead grid locator.
`what3Words`	`String`	What3Words address.
`bip39location`	`String`	BIP39 mnemonic encoding of the location.
`notes`	`String`	Freeform notes.
`codes`	`List<GeneratedCode>`	Generated codes for this location. See GeneratedCode.
`count`	`Integer`	Override code count (null = use CEOI default).
`addToWordlist`	`Boolean`	Override: include in word list (null = use CEOI setting).
`neededCount`	`int`	Effective code count after applying CEOI defaults.
`detail`	`String`	The preferred coordinate string based on the CEOI `locationDetailField` setting.
`mapFileName`	`String`	Name of associated map file, if any.
`mapFileResourceId`	`UUID`	Resource ID of the map file.

`localFile`	`File`	Runtime reference to the map image file.
`mapFileSize`	`long`	Map file size in bytes.

FreeMarker Usage Example

```
<#list CEOI.locations as loc>
${loc.name}: ${loc.mgrs} / ${loc.lat_lon}
  Codes:
  <#list loc.codes as c>
    P${c.period}D${c.division}: ${c.code}
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- Location / Document / Frequency Decrypt (Jasper)
- Location Codes — Encrypt
- Location Codes — Decrypt

FreeMarker Entity Reference: Frequency

Obtained from `CEOI.frequencies` — a `List<Frequency>`. Each `Frequency` represents a radio frequency with its mode, optional alternate name, and generated codes.

Frequency Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Frequency name.
`frequency`	`String`	Frequency value string (e.g., “146.520”).
`mode`	`String`	Transmission mode (e.g., “FM”, “USB”, “AM”).
`alternateName`	`String`	Alternate or net name for this frequency.
`description`	`String`	Returns `alternateName`.
`detail`	`String`	Returns the `frequency` value.
`notes`	`String`	Freeform notes.
`sortOrder`	`Integer`	Sort order index.
`range`	`int`	Transmission range (numeric value).
`units`	`DistanceUnits`	Distance units for the range (enum: e.g., `MILES`, `KILOMETERS`).
`excluded`	`boolean`	`true` if this frequency is excluded from code generation.
`codes`	`List<GeneratedCode>`	Generated codes for this frequency. See GeneratedCode.
`count`	`Integer`	Override code count (null = use CEOI default).
`addToWordlist`	`Boolean`	Override: include in word list (null = use CEOI setting).
`neededCount`	`int`	Effective code count after applying CEOI defaults.

FreeMarker Usage Example

```
<#list CEOI.frequencies as freq>
${freq.name}: ${freq.frequency} ${freq.mode}
```

```
  <#list freq.codes as c>
    P${c.period}D${c.division}: ${c.code}
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- UNIT_CALLSIGNS / FREQ_ENCODE (Jasper)
- Frequency Decrypt (Jasper)

FreeMarker Entity Reference: Document

Obtained from `CEOI.documents` — a `List<Document>`. Each `Document` is a named item (typically a file or publication) that receives generated code entries so operators can reference it securely in messages.

Document Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Document name.
`description`	`String`	Document description.
`fileName`	`String`	Associated file name (null if none attached).
`resourceId`	`UUID`	Resource identifier for the attached file.
`localFile`	`File`	Runtime local file reference.
`fileSize`	`long`	File size in bytes.
`notes`	`String`	Freeform notes.
`codes`	`List<GeneratedCode>`	Generated codes for this document. See GeneratedCode.
`count`	`Integer`	Override code count (null = use CEOI default).
`addToWordlist`	`Boolean`	Override: include in word list (null = use CEOI setting).
`neededCount`	`int`	Effective code count after applying CEOI defaults.
`linkId`	`UUID`	Returns `id` (word-list link interface).
`linkText`	`String`	Returns `name` (word-list link interface).

FreeMarker Usage Example

```
<#list CEOI.documents as doc>
${doc.name}${doc.description?has_content}: ${doc.description}}
  Codes:
  <#list doc.codes as c>P${c.period}D${c.division}: ${c.code}
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- Document / Location / Frequency Decrypt (Jasper)
- Document Codes — Encrypt
- Document Codes — Decrypt

FreeMarker Entity Reference: Word

Obtained from `CEOI.words` — a `List<Word>`. Each `Word` is an entry in the CEOI word list. It may stand alone or be linked to a Document, Location, or Frequency so that operators can encode that item by word.

Word Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`word`	`String`	The word text.
`name`	`String`	Returns `word`.
`description`	`String`	Returns `word`.
`codes`	`List<GeneratedCode>`	Generated codes for this word entry. See GeneratedCode.
`count`	`Integer`	Override code count (null = use CEOI default).
`neededCount`	`int`	Effective code count after applying CEOI defaults.
`linkId`	`UUID`	UUID of the linked Document, Location, or Frequency (null if `linkType == NONE`).
`linkType`	`WordLinkType`	Type of link: `DOCUMENT`, `LOCATION`, `FREQUENCY`, or `NONE`.

Resolving Linked Entities

Use the lookup methods on CEOI to resolve a word's linked entity:

```
<#if word.linkType == "DOCUMENT">
  <#assign linked = CEOI.getDocumentById(word.linkId)>
<#elseif word.linkType == "LOCATION">
  <#assign linked = CEOI.getLocationById(word.linkId)>
<#elseif word.linkType == "FREQUENCY">
  <#assign linked = CEOI.getFrequencyById(word.linkId)>
</#if>
```

Related Topics

- CEOI (Root Object)
- WORDLIST_ENCODE / WORDLIST_DECODE (Jasper)

- Wordlist Encode
- Wordlist Decode

FreeMarker Entity Reference: DRYAD

Obtained from `CEOI.cipherTables` — a `List<DRYAD>`. Each `DRYAD` is a 10×25 cipher substitution table assigned to a period and division. Its grid data is held in 25 `DRYADRow` objects accessible via `dryad.rows`.

DRYAD Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Table name.
`period`	`Integer`	Period this table belongs to (1-based).
`division`	`int`	Division within the period (1-based).
`number`	`Integer`	Sequential number within the division.
`rows`	`List<DRYADRow>`	The 25 rows of the cipher grid. See DRYADRow below.
`chars`	`char[]`	All table characters as a flat char array.
`printLines`	`List<String>`	Pre-formatted rows suitable for plain-text printing.
`SHA1`	`String`	SHA-1 hash of the table data.
`SHA256`	`String`	SHA-256 hash.
`SHA512`	`String`	SHA-512 hash.
`assignmentList`	`AssignedAssetList`	Units and groups assigned this DRYAD.
`assetName`	`String`	Formatted identifier: period, division, and number.

Sub-Object: DRYADRow (from `dryad.rows`)

One of the 25 rows of the DRYAD grid. Each row has a one-character label and 10 cell values. Access cells by index using the method call syntax: `${row.getCol(n)}`.

Property / Call	Java Type	Description
`row`	`String`	Row label character (typically A–Z minus a few letters).
`col0`	`String`	Cell value for column 0 (digit 0).
`col1`	`String`	Cell value for column 1.

`col2` – `col9`	`String`	Cell values for columns 2–9.
`getCol(n)`	`String`	Get column value by index (0–9). Call as `${row.getCol(3)}`.

FreeMarker Usage Example

```
<#list CEOI.cipherTables as dryad>
${dryad.name} (Period ${dryad.period}, Division ${dryad.division})
SHA-1: ${dryad.SHA1}

  <#list dryad.rows as row>
${row.row} | ${row.col0} ${row.col1} ${row.col2} ${row.col3} ${row.col4}
           ${row.col5} ${row.col6} ${row.col7} ${row.col8} ${row.col9}
  </#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- CYPHERTABLES (DRYAD) Data Source (Jasper)
- DRYADs Report
- DRYADs

FreeMarker Entity Reference: Sarneg

Obtained from `CEOI.sarnegs` — a `List<Sarneg>`. Each `Sarneg` is a 10-character keyword used as a substitution grid header for encoding alphanumeric values. Individual letters are accessible by index.

Sarneg Properties

Property / Call	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	SARNEG name.
`word`	`String`	The 10-character SARNEG keyword.
`period`	`int`	Period this SARNEG belongs to (1-based).
`division`	`int`	Division within the period (1-based).
`number`	`int`	Sequential number within the division.
`SHA1`	`String`	SHA-1 hash of the keyword.
`assignmentList`	`AssetAssignmentList`	Units and groups assigned this SARNEG.
`assetName`	`String`	Formatted identifier with period, division, and number.
`getLetter(i)`	`String`	Character at position `i` (0–9). Call as `${sarneg.getLetter(0)}`.

> The ten letters of the SARNEG keyword form the column headers of the substitution grid. Letter 0 maps to digit 0, letter 1 to digit 1, and so on through letter 9.

FreeMarker Usage Example

```
<#list CEOI.sarnegs as sn>
${sn.name} (P${sn.period} D${sn.division} #${sn.number})
Keyword: ${sn.word}
Columns: 0=${sn.getLetter(0)} 1=${sn.getLetter(1)} ... 9=${sn.getLetter(9)}
</#list>
```

Related Topics

- CEOI (Root Object)
- SARNEG Data Source (Jasper)
- SARNEGs Report

- SARNEGs Panel

FreeMarker Entity Reference: ChallengeResponse

Obtained from `CEOI.challengeResponse` — a `List<ChallengeResponse>`. Each `ChallengeResponse` holds a triplet of words: a challenge word that is issued, a response word that proves identity, and a duress word that signals coercion.

ChallengeResponse Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`name`	`String`	Display name, or formatted “Period: X, Division: Y, #: Z” if none set.
`period`	`Integer`	Period this pair belongs to (1-based).
`division`	`Integer`	Division within the period (1-based).
`number`	`int`	Sequential number within the division.
`challengeWord`	`String`	The word issued as a challenge.
`responseWord`	`String`	The correct response proving identity.
`duressWord`	`String`	The response word used under coercion to signal distress.
`SHA1`	`String`	SHA-1 hash of the word set.
`assignmentList`	`AssetAssignmentList`	Units and groups assigned this challenge pair.
`assetName`	`String`	Formatted identifier with period, division, and number.

FreeMarker Usage Example

```
<#list CEOI.challengeResponse as cr>
P${cr.period} D${cr.division} #${cr.number}
  Challenge: ${cr.challengeWord}
  Response:  ${cr.responseWord}
  Duress:    ${cr.duressWord}
</#list>
```

Related Topics

- CEOI (Root Object)
- CHALLENGE_RESPONSE Data Source (Jasper)
- Challenge Response Words Report

- Challenge/Response Panel

FreeMarker Entity Reference: OneTimePad & Codex

One-time pads are obtained from `CEOI.oneTimePads` — a `List<OneTimePad>`. Codex lookup tables are obtained from `CEOI.codexList` — a `List<Codex>`.

OneTimePad Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`period`	`int`	Period this pad belongs to (1-based).
`division`	`int`	Division within the period (1-based).
`number`	`int`	Sequential number within the division.
`type`	`OTPType`	OTP character type: `NUMBERS`, `DIANA`, or `DIANA_PLUS`.
`padId`	`String`	Pad series/identifier string (from the pad's series name field).
`keyId`	`String`	The next 5 characters from the current pad position (key ID for the pad sheet header).
`size`	`int`	Total number of characters in the pad.
`remaining`	`int`	Number of unused characters remaining.
`startPosition`	`int`	Index of the first unused character (−1 if pad is exhausted).
`chars`	`char[]`	Raw pad data as a char array.
`imported`	`boolean`	`true` if this pad was imported from an external source.
`SHA1`	`String`	SHA-1 hash of the pad data.
`SHA256`	`String`	SHA-256 hash.
`SHA512`	`String`	SHA-512 hash.
`assignmentList`	`AssetAssignmentList`	Units and groups assigned this pad.
`assetName`	`String`	Formatted identifier with period, division, and number.

Codex Properties

A `Codex` is a conversion table that maps characters to integers (and back), used alongside one-time pads.

Property / Call	Java Type	Description
`name`	`String`	Codex name (e.g., “DIANA”, “Numbers”).
`entries`	`Collection<CodexEntry>`	All entries in this codex. See CodexEntry.
`validChars`	`String`	String of all valid characters in this codex.
`getNumberFor(letter)`	`int`	Returns the numeric value for the given letter. Call as `${codex.getNumberFor("A")}`.
`getValueFor(number)`	`String`	Returns the character(s) for the given numeric index.
`getLetterFor(number)`	`String`	Returns the raw letter value for the given number (may differ from `getValueFor` for meta entries).

Sub-Object: CodexEntry (from `codex.entries`)

Field	Java Type	Description
`number`	`int`	Numeric key for this entry.
`letter`	`String`	Character or meta value (e.g., `"SPC"` for space, `"REQ"` for required).

FreeMarker Usage Example

```
<#list CEOI.oneTimePads as otp>
P${otp.period} D${otp.division} #${otp.number} — ${otp.padId}
Type: ${otp.type}  Size: ${otp.size}  SHA-1: ${otp.SHA1}
</#list>

<#list CEOI.codexList as codex>
Codex: ${codex.name}  Valid chars: ${codex.validChars}
</#list>
```

Related Topics

- CEOI (Root Object)

- ONE_TIME_PAD Data Source (Jasper)
- OTP Codex Data Source (Jasper)
- One Time Pad Report
- One Time Pads Panel

FreeMarker Entity Reference: SOI

Obtained from `CEOI.SOIs` — a `List<SOI>`. Each `SOI` (Signals Operating Instruction) defines PACE frequencies, unit assignments, a comm schedule, and an optional SARNEG for a specific operational net.

SOI Properties

Property	Java Type	Description
`id`	`UUID`	Unique identifier.
`title`	`String`	SOI title.
`name`	`String`	Returns `title`.
`objective`	`String`	Operational objective text.
`duration`	`int`	Duration of the SOI in time units.
`locationId`	`UUID`	Primary location UUID. Resolve with `CEOI.getLocationById(soi.locationId)`.
`duressCode`	`String`	Duress signal code.
`controlUnitId`	`UUID`	Net control unit UUID. Resolve with `CEOI.getUnitById(soi.controlUnitId)`.
`controlUnitCallSign`	`String`	Net control unit call sign.
`recoveryUnitId`	`UUID`	Recovery unit UUID.
`recoveryUnitCallSign`	`String`	Recovery unit call sign.
`sarneg`	`Sarneg`	Associated SARNEG object. May be null. See Sarneg.
`unitsAssignments`	`List<SOIUnitAssignment>`	Unit assignments and roles. See SOIUnitAssignment.
`commSchedule`	`List<CommScheduleItem>`	Communication schedule (up to 10 items). See CommScheduleItem.
`assignmentList`	`AssetAssignmentList`	Units and groups assigned this SOI.

PACE Frequency IDs

Each PACE entry stores the UUID of a `Frequency` object. Resolve with `CEOI.getFrequencyById(id)`.

Property	Java Type	Description

`primaryTxFrequencyId`	`UUID`	Primary transmit frequency.
`primaryRxFrequencyId`	`UUID`	Primary receive frequency (null = same as Tx).
`primarySignal`	`String`	Primary signal / authentication note.
`primaryProword`	`String`	Primary proword.
`alternateTxFrequencyId`	`UUID`	Alternate transmit frequency.
`alternateRxFrequencyId`	`UUID`	Alternate receive frequency.
`alternateSignal`	`String`	Alternate signal note.
`alternateProword`	`String`	Alternate proword.
`contingencyTxFrequencyId`	`UUID`	Contingency transmit frequency.
`contingencyRxFrequencyId`	`UUID`	Contingency receive frequency.
`contingencyProword`	`String`	Contingency proword.
`emergencySignal`	`String`	Emergency signal note.
`emergencyProword`	`String`	Emergency proword.

Sub-Object: SOIUnitAssignment (from `soi.unitsAssignments`)

Property	Java Type	Description
`unitId`	`UUID`	UUID of the assigned unit. Resolve with `CEOI.getUnitById(a.unitId)`.
`unit`	`Unit`	Pre-resolved `Unit` object (available at runtime). See Unit.
`role`	`String`	Unit's role in this SOI (e.g., "Net Member", "Net Control").
`callSign`	`String`	Call sign for this assignment.

Sub-Object: CommScheduleItem (from `soi.commSchedule`)

One day of the communication schedule. Each day has a list of `CommWindow` time windows.

Property	Java Type	Description
`windows`	`List<CommWindow>`	Communication windows for this day. See below.

CommWindow (from `item.windows`)

Property	Java Type	Description
`startTime`	`int`	Start time in HHMM format (e.g., `1430` = 14:30).
`endTime`	`int`	End time in HHMM format.
`primaryFrequencyId`	`UUID`	Primary frequency UUID. Resolve with `CEOI.getFrequencyById(w.primaryFrequencyId)`.
`altFrequencyId`	`UUID`	Alternate frequency UUID. May be null.

FreeMarker Usage Example

```
<#list CEOI.SOIs as soi>
${soi.title}
Control: ${soi.controlUnitCallSign}
Primary Freq: <#if soi.primaryTxFrequencyId??>${CEOI.getFrequencyById(soi.primaryTxFr

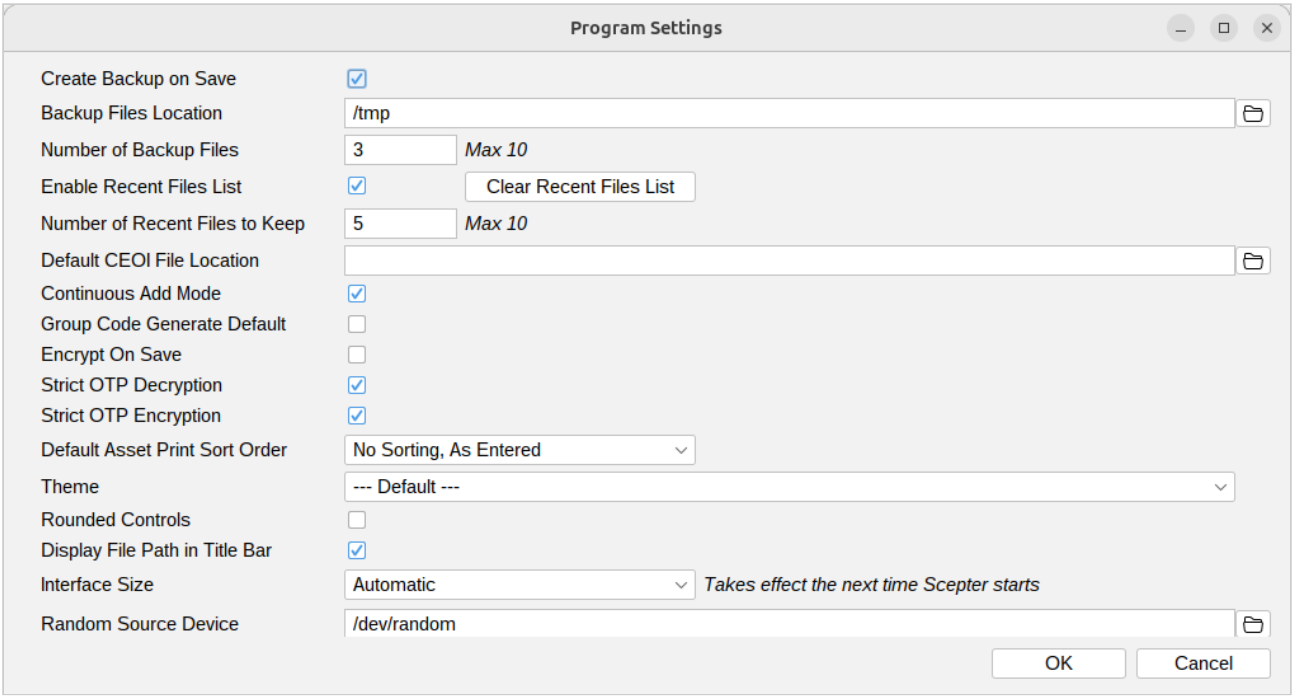
Units:
<#list soi.unitsAssignments as a>
  ${a.unit.name} / ${a.callSign} (${a.role})
</#list>

Comm Schedule:
<#list soi.commSchedule as day>
  Day ${day?index + 1}:
  <#list day.windows as w>
    ${w.startTime?string("0000")}–${w.endTime?string("0000")}
  </#list>
</#list>
</#list>
```

Related Topics

- CEOI (Root Object)
- SOI Data Source (Jasper)
- SOI Report
- Signals Operating Instructions
- SOI Editor

Program Settings



Open it from **File** → **Settings** → **Preferences ...**. The dialog is titled Program Settings, and controls global application preferences. Changes take effect when you click **OK**. Click **Cancel** to discard all changes.

Backup Settings

Setting	Description
Create Backup on Save	When checked, Scepter automatically creates a backup copy of your CEOI file each time it is saved.
Backup Files Location	The directory where backup files are saved. Click the folder icon to browse.
Number of Backup Files	The maximum number of backup files to keep (up to 10). Older backups are deleted when the limit is reached.

Recent Files

Setting	Description
Enable Recent Files List	Shows recently opened CEOI files in the File menu for quick access.
Clear Recent Files List	Immediately clears all entries from the recent files list.
Number of Recent Files to Keep	Maximum number of recent files to track (up to 10).

File Locations

Setting	Description
Default CEOI File Location	The default directory used when opening or saving CEOI files. Click the folder icon to browse.

General

Setting	Description
Continuous Add Mode	When enabled, after adding a new entity (unit, group, etc.) the add dialog reopens automatically for rapid data entry.
Group Code Generate Default	Sets the default code generation behavior for groups.
Default Asset Print Sort Order	The default sort order used when opening the Asset Print dialog.

Security

Setting	Description
Encrypt On Save	When enabled, CEOI files are encrypted automatically when saved.
Strict OTP Decryption	When enabled, OTP decryption requires a perfect pad match. Relaxing this allows partial matches.
Strict OTP Encryption	When enabled, OTP encryption enforces strict pad length requirements.

Appearance

Setting	Description
Theme	Select the application look and feel. Changes apply after restarting Scepter.
Rounded Controls	When checked, buttons and input fields use rounded corners.
Display File Path in Title Bar	When checked, the title bar shows the full path of the open CEOI file. Uncheck it to show only the file name, which is worth doing before demonstrating Scepter to others or recording the screen, since the full path names the folders it is stored in.
Interface Size	Magnifies the whole interface on a high resolution screen. **Automatic** takes the magnification from the desktop. Choose a percentage to override it. Changes apply after restarting Scepter.

> On a high resolution screen Scepter can start up very small, because Java takes its magnification from the desktop's whole-number scaling factor and ignores fractional scaling. If that happens, set **Interface Size** to 200% and restart. Only whole multiples work: the setting offers 100%, 200%, 300% and 400% for that reason. For a single run without changing the setting, start Scepter with `SCEPTER_UI_SCALE=2 bin/scepter`.

Random Source Device

Specifies the OS random device used for cryptographic operations (e.g., `/dev/random` or `/dev/urandom`). The device must exist and be readable. Click the folder icon to browse. See Random Number Generation for a full explanation of how Scepter uses this device and guidance on choosing the right value.

Related Topics

- Main Window
- Set File Password
- Random Number Generation

Random Number Generation

Scepter generates all cryptographic material — callsigns, authentication codes, passwords, OTP pads, DRYAD tables, SARNEGs, and challenge/response pairs — using a random number generator that combines two independent entropy sources. Understanding this design helps you choose the right configuration for your operating environment.

How It Works

Each time Scepter needs a block of random bytes it performs the following steps:

1. Java's built-in **SecureRandom** generator (seeded from the JVM's entropy pool at startup) fills a 4 096-byte buffer.
2. If the configured **Random Source Device** file exists and is readable, Scepter reads 4 096 bytes from that device and adds them to the mix.
3. Both buffers are fed into a **SHA-512** hash. The resulting 64-byte digest becomes the output block.
4. Bytes are drawn from that 64-byte block until it is exhausted, then the process repeats.

Because the two sources are combined by hashing, the output is at least as strong as the stronger of the two. If the device file is unavailable, Scepter falls back to `SecureRandom` alone and continues without error.

> Scepter never writes the raw output of either source directly into crypto material; all output passes through SHA-512 first. This means even a weak or slow device file does not reduce the overall cryptographic strength below that of `SecureRandom`.

Random Source Device Options (Linux)

Device	Behavior
`/dev/urandom`	Non-blocking. Draws from the kernel's CSPRNG. Recommended for most users and environments, including Tails. Never stalls generation even when the entropy pool is low.
`/dev/random`	Blocks when the kernel's entropy pool runs low. On modern Linux kernels (5.6+) this behaves identically to `/dev/urandom` and is equally safe.
`/dev/hwrng`	Direct hardware random number generator (e.g., an on-chip TRNG or TPM). Provides true hardware entropy if the device is present. Not available on all systems.
Any readable file or device	Scepter treats the path as a byte stream and reads from it. This allows custom entropy sources such as audio devices or external hardware.

> On Tails, `/dev/urandom` is the recommended setting. Tails seeds the kernel entropy pool from multiple sources at boot, so `/dev/urandom` provides strong randomness from the moment Scepter starts.

Changing the Random Source Device

1. Open **File** → **Settings** → **Preferences ...**.
2. Scroll to the **Random Source Device** field near the bottom of the dialog.
3. Type the device path directly, or click the folder icon to browse to it.
4. Click **OK**. Scepter validates that the path exists and is readable before saving. If it is not, an error message is shown and the change is not applied.

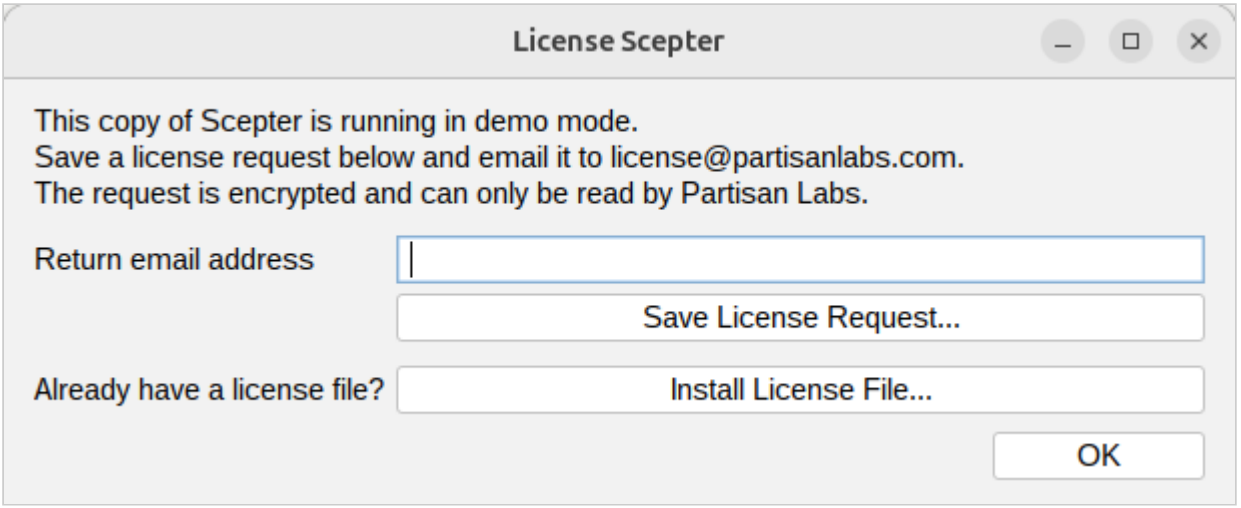
The new device takes effect immediately for all subsequent code generation in the same session.

> Setting the device to a non-random source such as `/dev/zero` (all zeros) or a static file reduces the entropy contributed by that source to zero. Scepter will still generate output using `SecureRandom`, but best practice is to always configure a genuine random device for operational use.

Related Topics

- Program Settings
- Generated Code Patterns
- Generation Modes

License



The License dialog is how an unlicensed copy of Scepter becomes a licensed one. It writes a license request for you to email to Partisan Labs, and installs the license file that comes back.

Open it from **Help** → **License...**. The menu item only appears when the copy you are running is unlicensed, so on a stick bought with Scepter preinstalled you will never see it.

Demo Mode

An unlicensed copy runs in demo mode, and the title bar ends with **DEMO**. You can open a CEOI, look through it, edit it and print it, but these are grayed out:

- **File > Save** and **File > Save As**. Answering **Yes** to "save changes?" when you exit, start a new CEOI or install an update does not save either; Scepter says the file was not saved.
- **Actions > Generate Crypto...**, so no new codes are generated.
- Clearing codes: **Actions > Remove Generated Crypto...**, **Clear ALL Codes** in the unit editor, and **Delete Selected** and **Clear All** on a codes tab.

The example CEOIs that ship with Scepter already hold generated codes, so they show what the program produces.

> Demo mode is a good way to try Scepter before buying, and it is also what you see if the license file is missing or is not a genuine Scepter license.

Invalid License

A genuine license is issued for one computer's drive. If the installed license was issued for a different computer (or for a drive that is not attached), Scepter does not go into demo mode. Instead it shows a **License Violation** notice for one minute when it starts. The notice cannot be closed early. When it closes, Scepter runs normally, with saving enabled, and the title bar ends with **Invalid License** instead of DEMO. The License dialog then opens over the main window so you can request a license for this computer.

The same one-minute notice appears before every save, including **Save As** and saving when you are asked to save changes; the file is saved when the notice closes. Install a license issued for this computer to stop it.

Requesting a License

Field	Description
Order number	The number of your Partisan Labs order, from the order confirmation email, such as 1424. Required — it is how the request is matched to your order.
Return email address	Where the license file will be sent back to. Required — the request cannot be saved without one.
Save License Request...	Writes the request to a file you choose.

1. Type your order number, and the email address the license should be sent back to.
2. Click **Save License Request...** and choose where to write the file. It is named after your order and the date, such as `scepter-license-request-1424-20261103.asc`.
3. Email that file to the address shown in the dialog.

The request is encrypted and can only be read by Partisan Labs. It identifies the machine so that the license can be issued for it, and it carries the order number and return address you typed — nothing else.

The license is tied to one drive. On Tails that is your Tails USB stick, which Scepter finds through the Persistent Storage it is installed in; on another system it is the disk the system started from. Remove any other USB sticks before you save the request, so it is clear which drive is yours.

Installing a License

When the license file arrives, click **Install License File...** and select it. Scepter checks the file before installing it, and may ask you one of two questions first.

Question	What it means
License Does Not Match	Either the file is not a genuine Scepter license (installed anyway, Scepter stays in demo mode), or it is a genuine license issued for a different computer or for a drive that is not attached right now. If it was issued for a drive, attach it and try again. Installed anyway on the wrong computer, Scepter runs with the Invalid License notice described above.
Replace License	A license is already installed. Answering yes overwrites it with the new one.

> Restart Scepter after installing a license. The running copy stays in demo mode until you do.

Which License You Have

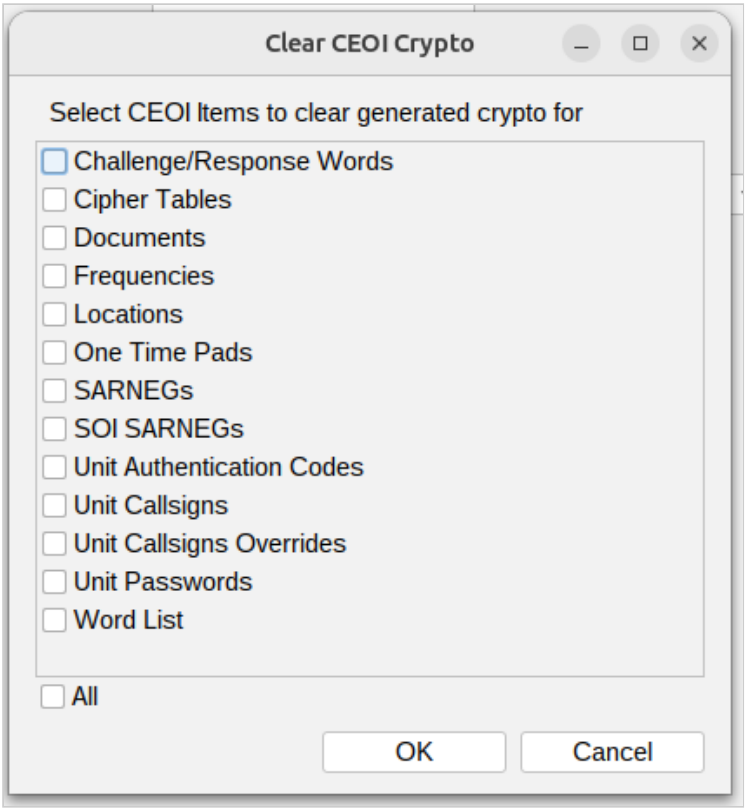
Every license has a short ID — four groups of four characters, such as `A582-81E8-B555-7EB0`. It is shown on the splash screen at startup and in **Help** → **About**, where a button beside it copies it to the clipboard so you can paste it into an email.

> Quote that ID when you write to Partisan Labs about a license. It is how your license is found without you having to describe your machine or dig out the original email.

Related Topics

- Main Window

Clear CEOI Crypto



The Clear CEOI Crypto dialog lets you selectively clear (erase) generated cryptographic material from the current CEOI. Use this when you need to regenerate specific types of codes without starting the entire CEOI over from scratch.

> Clearing cryptographic material is irreversible. Any cleared codes must be regenerated before the CEOI can be used operationally. Ensure you have a backup before clearing.

Selecting What to Clear

A checkbox is shown for each type of coded item in the CEOI (such as DRYADs, SARNEGs, challenge/response pairs, OTPs, word lists, and frequencies). Check the types you want to erase.

Codes you typed in yourself (overrides) are cleared separately. **Unit Callsigns**, **Frequencies**, **Locations** and **Documents** clear only the generated codes and keep the overrides. To clear the overrides, check **Unit Callsigns Overrides**, **Frequency Code Overrides**, **Location Code Overrides** or **Document Code Overrides**.

Use the **Select All / Deselect All** checkbox at the bottom to quickly toggle all items on or off.

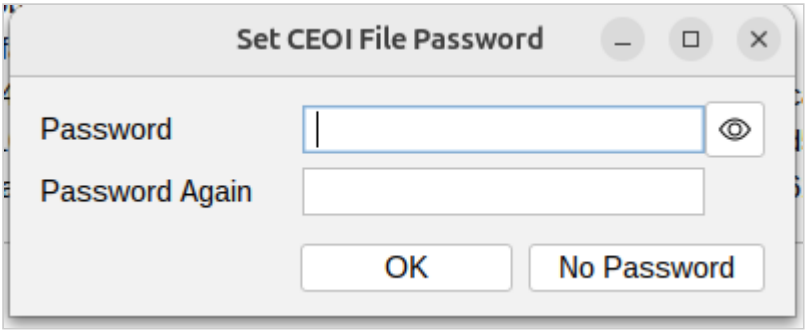
Clearing

Click **OK** to clear the selected types. Click **Cancel** to close the dialog without making any changes.

Related Topics

- Main Window
- Set File Password
- CEOI Defaults

Set File Password



The Set File Password dialog lets you create or change the password used to encrypt the current CEOI file. The password must be entered twice to confirm it. Once set, the file will be encrypted with this password whenever it is saved.

Fields

Field	Description
Password	Enter the new password. The password is masked by default.
Confirm Password	Re-enter the same password to confirm. Both fields must match.

Show / Hide Password

Click the eye icon next to the Password field to toggle between showing and masking the password in both fields simultaneously.

Saving the Password

Click **OK** to set the password. If the two fields do not match, or if the password is empty, an error message is shown and you must correct the input before proceeding.

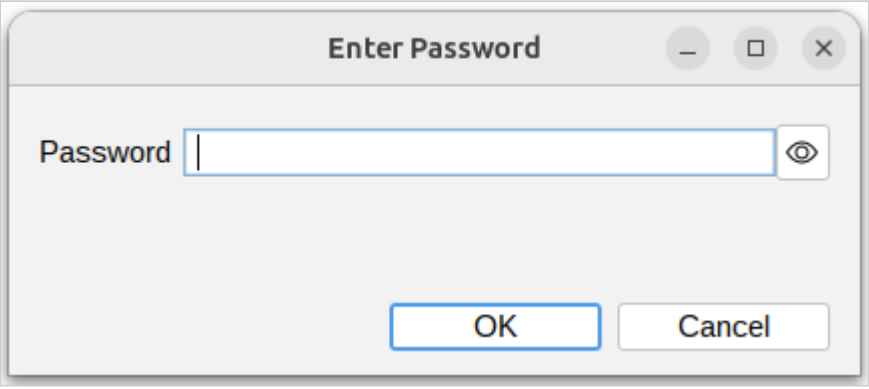
Click **Cancel** to close the dialog without changing the password.

> If you forget the file password, you will not be able to open the encrypted CEOI file. Keep a secure record of your password.

Related Topics

- Password Entry Dialog
- Program Settings

Password Entry



The Password Entry dialog prompts you to enter the password needed to open an encrypted CEOI file. It is displayed automatically when Scepter attempts to open a password-protected file.

Fields

Field	Description
Password	Enter the password for the encrypted file. The password is masked by default.

Show / Hide Password

Click the eye icon to toggle between showing and masking the entered password.

Submitting

Click **OK** or press **Enter** to submit the password. If the password is incorrect, Scepter will display an error and you will need to try again.

Click **Cancel** to cancel opening the file.

Related Topics

- Set File Password
- Program Settings