

DotTalk++ / x64base Developer Manual

Assembled build — candidate artifact

2026-07-20

DotTalk++ / x64base Developer Manual

Assembled artifact — produced by the manifest-driven assembler from the declared bill of materials.
This is a candidate build; acceptance is gated.

Manual id	<code>developer_manual_publication_v1</code>
Public source snapshot	<code>8ee746dee</code>
Accepted reader baseline	<code>EA2E12A9D3E1</code> (accepted 2026-07-18T05:06:03Z)
Command-reference pages	183
Build date (UTC)	2026-07-20T19:17:26Z
Machine (maintainer-attested)	Alienware m16 R2 / Intel Core Ultra 9 185H

Provenance & attestation

This manual is assembled from source, HELP/metadata, SelfDoc reports, and reviewed manualgen sections. Proof labels travel with each part.

- Accepted reader baseline: `EA2E12A9D3E1`, 4118 lines, 237 headings (2026-07-18T05:06:03Z).
- Command reference: 183 pages (164 reader-linked + 19 supplemental).
- MDO lane: `D0CFLUSH-20260716-001`.
- Machine attestation: `MAINTAINER_ATTESTED` (Alienware m16 R2). The individual run transcripts did not self-record the machine; this binding is a maintainer attestation made after the fact, not a value captured inside each run's output. Recorded as `MAINTAINER_ATTESTED` rather than silently auto-attached.

Preface — how to read this manual

To be authored.

Table of Contents

- Provenance & attestation
- Preface — how to read this manual
- Command Reference
 - `ALLTRIM`
 - `APPEND`
 - `AREA`
 - `AREA51`
 - `ASC`
 - `ASCEND`
 - `ASCII`
 - `AT`
 - `ATC`
 - `AUTODBF`

- BIBLETALK
- BOTTOM
- BROWSE
- BROWSER
- BUFFERING
- BUILDLMDB
- CANARY
- CDX
- CHR
- CHRISTMAS
- CLOSE
- CMDREL
- CNX
- COBOL
- CODASYL
- COMMANDSHELP
- COMMIT
- CONCAT
- CONTINUE
- COPY
- CREATE
- CTOD
- DATE
- DECISION
- DELETE
- DESCEND
- DIR
- DO
- DOTHHELP
- DOTSCRIPT
- DRAWIO
- DTOC
- DUMP
- EDIT
- EDUCATIONAL_USE
- ELSE
- ENDIF
- ENDLOOP
- ENDSCAN
- ENDUNTIL
- ENDWHILE
- ERASE
- ERP
- ERSATZ
- EVAL
- EVALUATE
- EXAMPLE
- EXPFUNCS
- EXPORT
- EXPORTSQL
- EXPRESSION
- FIND
- FOXHELP
- FOXPRO
- FOXSTANDARD
- FOXTALK

- GLOSSARY
- GO
- GOTO
- GPS
- IDX
- IF
- IMAGE
- IMPORT
- IMPORTSQL
- INDEX
- INIT
- INTRO
- LEFT
- LEN
- LIST
- LMDB
- LMDB_UTIL
- LMDBDUMP
- LOCATE
- LOCK
- LOOP
- LOOP_BUFFER
- LOOPS
- LOWER
- LTRIM
- MCC
- MODEL
- NAVIGATION
- NORMALIZE
- ORDER
- PACK
- PADC
- PADL
- PADR
- PREDHELP
- PREDICATE
- PREDICATES
- PROJECTION
- PROJECTS
- PROPER
- PSHELL
- REBUILD
- RECALL
- REINDEX
- REL
- REL_ENUM
- REL_REFRESH
- RELATION
- RELATIONS
- REPLACE
- REPLICATE
- RETRO
- RIGHT
- ROLLBACK
- RTRIM
- RULE

- RUN
- SCAN
- SCAN_BUFFER
- SCHEMAS
- SCRIPT
- SCX
- SECHO
- SECURITY
- SEEK
- SELECT
- SFTP
- SHELL
- SHOWINI
- SHUTDOWN
- SIX
- SKIP
- SMARTLIST
- SPACE
- SQL
- SQLERASE
- SQLHELP
- SQLITE
- SQLSEL
- SQLVER
- STATE
- STR
- STRUCT
- STU_REPEAT
- STU_UPPER
- STUDENTECHO
- STUDENTHELLO
- STUFF
- SUBSTR
- TABLE_BUFFER
- TEXT
- TIME
- TOP
- TRIM
- TUPEXPORT
- TUPLE
- TUPLEDELTA
- TUPTALK
- TUPVALIDATE
- TURBOPACK
- TVISION
- UNDELETE
- UNLOCK
- UNTIL
- UNTIL_BUFFER
- UPDATE
- UPPER
- USE
- VAL
- VALIDATE
- WEB
- WHILE

- WHILE_BUFFER
 - WORKSPACE
 - WSREPORT
 - ZAP
 - ZIP
- Function Reference
- Appendix: Review and Deferred: SET-family
 - Commands in this section
 - Notes for future prose pass
- Error / Message Catalog
- Tables, Records, and Data Model
 - Purpose of this section
 - Source-lane rule for this section
 - Tables, records, and fields
 - Schema and DDL
 - Record views and memo fields
 - Work areas as context, not the main topic
 - Compatibility and bridge material
 - Command and concept map
 - What this section should not do yet
 - Review notes before candidate generation
 - Boundary
- Command Surface, Dispatch, and Entry Variants
 - Evidence lanes
 - Authority boundaries
 - Command surface
 - Canonical commands
 - Subcommands and command families
 - Aliases and entry variants
 - Parser dispatch and handlers
 - Function bridge behavior
 - AGGS boundary
 - SET family boundary
 - Generated command pages
 - HELP and CMDHELPCHK
 - Future META alignment
 - Slow-lane canary tracking names
 - Review notes before PIP-003
- Expressions, Querying, and Aggregates
 - Expression evaluation surfaces
 - CALC and CALCWRITE
 - Predicates, WHERE, and FOR
 - LOCATE, CONTINUE, and SCAN
 - COUNT and aggregate commands
 - Direct aggregate verbs and AGGS
 - Scalar functions versus aggregate commands
 - Function command-line bridge
 - Deleted-record filters
 - Error and null behavior
 - HELP FUNCTIONS and FUNCTION help
- Indexing, Tags, Relations, and Views
 - Indexing vocabulary
 - Open index architecture note
 - Logical order and active order
 - Ownership reminder
 - Tags and tag availability

- CDX, CNX, and LMDB boundary
- SET-family boundary and canonicalization canary
- SEEK and FIND active-order boundary
- Reindexing and rebuild behavior
- Relations and relation traversal
- Views and projection boundary
- ERSATZ and browser caution
- Known canaries
- Messages, Errors, and Diagnostics
 - Message vocabulary
 - Typed message catalog direction
 - SYSMSG and SYSTEM_MESSAGES
 - HELP rows as diagnostic evidence
 - SHARED_MSG caution
 - CMDHELPCHK role
 - Runtime diagnostic examples
 - Severity vocabulary
 - Parser warnings and syntax diagnostics
 - No-active-table and not-found messages
 - Message catalog and HELP alignment
- HELP, Metadata, CMDHELPCHK, and Manualgen Alignment
 - Core doctrine
 - Assembly workflow versus truth authority
 - HELP lane
 - Metadata lane
 - CMDHELPCHK lane
 - SelfDoc lane
 - Manualgen lane
 - Source lane
 - Runtime lane
 - Temporary evidence lanes
 - Crosswalk discipline
 - Safety boundaries
- Command Reference Assembly, Aliases, and Generated Page Hygiene
 - Authority model for command reference assembly
 - Duplicate command rows
 - Aliases and variants
 - Canonical command identity
 - Slug collisions
 - APPEND BLANK and APPEND_BLANK
 - LOAD guard
 - SET-family canonicalization
 - Internal owner and public surface
 - Metadata feeders
 - Command-reference crosswalks
 - Publication readiness
 - No-delete and no-mutation safety
- Promoted Draft Review, Header Normalization, and Publication Readiness
 - Promoted draft workspace
 - Inspectable files
 - Review packets and inspection packets
 - Candidate note headers
 - Reviewed candidate status blocks
 - Header normalization
 - Substantive prose boundary
 - Path repair and canonical path verification

- Slugs and section ids
- Section count
- Section order
- Final publication boundary
- Generated command pages and evidence artifacts
- HELP, CMDHELPCHK, and metadata boundaries
- Report-only publication readiness
- Human review
- Runtime Evidence, Source Verification, and Canary Closure
 - Evidence doctrine in practice
 - Runtime evidence
 - Source verification
 - Runtime and source together
 - Canary lifecycle
 - Opening a canary
 - Keeping canaries visible
 - Closing a canary
 - Legacy documents
 - Smoke tests, shakedown, regressions, builds, and releases
 - REGRESSION and TEST as proof launchers
 - HELP, CMDHELPCHK, and metadata in evidence practice
 - SelfDoc and manualgen evidence
 - Evidence crosswalks
 - No-mutation safety
 - Canary non-disappearance boundary
- Diagrams
- Appendices
- Glossary
- Colophon — build provenance

Command Reference

183 command pages bound from the reference set.

ALLTRIM

ALLTRIM

- Catalog/topic: **FOX / ALLTRIM**
- Status: **supported**
- Implemented/supported: **T / T**
- Primary/confidence: **FOXREF / CATALOG**

Summary Remove leading and trailing spaces from <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- ALLTRIM(<cExpression>)

Provenance

- Topic key: **FOX|ALLTRIM**
- Included HELP rows: **3**
- HELP reference run: **MANRUN-20260717T222026Z-28C704E0**

- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

APPEND

APPEND

- Catalog/topic: DOT / APPEND
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Append a new record using current table defaults and active buffering rules.

- Append one or more blank records to the current table, using smart append paths that maintain keys and active indexes, or raw append paths when requested.

Status

- implemented=yes; supported=yes

Syntax

- APPEND USAGE
- APPEND
- APPEND <count>
- APPEND MANY <count>
- APPEND RAW
- APPEND RAW MANY <count>

Usage

- APPEND USAGE
- APPEND
- APPEND <count>
- APPEND MANY <count>
- APPEND RAW
- APPEND RAW MANY <count>

Note

- APPEND with no arguments appends one blank record through the shared smart append path.
- APPEND <count> is shorthand for APPEND MANY <count>.
- APPEND MANY <count> performs smart batch append under one lock.
- APPEND RAW appends one record without inline index update.
- APPEND RAW MANY <count> performs raw batch append under one lock.
- Count values must be positive integers.
- APPEND is a table-data mutation command; do not classify it as read-only.
- risk:
- writes_dbf_records: yes
- appends_records: yes
- updates_indexes: smart append paths
- raw_path_skips_inline_index_update: yes
- one_lock_batch: MANY forms
- requires_open_table: yes

Related

- APPEND_BLANK
- REPLACE
- MULTIREP
- TABLE
- COMMIT

Provenance

- Topic key: DOT|APPEND
- Included HELP rows: 34
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

AREA

AREA

- Catalog/topic: DOT / AREA
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Report the current DotTalk++ work-area state.

- Report the current work-area slot and table context
- Report the current work-area slot and current area file/session state.

Status

- implemented=yes; supported=yes

Syntax

- AREA
- AREA USAGE

Usage

- AREA
- AREA USAGE

Example

- AREA

Note

- This is a non-mutating inspection command
- Use SELECT to change the current work area
- AREA with no arguments reports the current work-area number, open file, record count, current record, DBF flavor, runtime kind, logical name, absolute path, and active order/index line.
- AREA is read-only; it reports current area state and does not mutate table data.

Related

- DBAREA
- DBAREAS
- STATUS
- STRUCT
- WORKSPACE

Provenance

- Topic key: DOT|AREA
- Included HELP rows: 20
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

AREA51

AREA51

- Catalog/topic: DOT / AREA51
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer sandbox / experimental command.

Status

- implemented=yes; supported=yes

Syntax

- AREA51

Provenance

- Topic key: DOT|AREA51
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ASC

ASC

- Catalog/topic: FOX / ASC
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the ASCII code (0-255) of the first character of <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- ASC(<cExpression>)

Provenance

- Topic key: FOX|ASC
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ASCEND

ASCEND

- Catalog/topic: DOT / ASCEND
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Set ascending sort direction for the active order/tag.

- Set the active order/tag direction to ascending for the current work area.

Status

- implemented=yes; supported=yes

Syntax

- ASCEND
- ASCEND USAGE

Usage

- ASCEND
- ASCEND USAGE

Note

- ASCEND requires an active order except for ASCEND USAGE.
- ASCEND with no arguments mutates order direction to ascending.
- ASCEND does not mutate table records or rebuild indexes.
- risk:
- mutates_order_state: yes
- mutates_table_data: no
- requires_active_order: yes except usage

Related

- DESCEND
- SET ORDER
- ORDER
- include “xbase.hpp”

Provenance

- Topic key: DOT|ASCEND
- Included HELP rows: 18
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ASCII

ASCII

- Catalog/topic: DOT / ASCII
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display ASCII/character-code reference information for teaching and diagnostics.

Status

- implemented=yes; supported=yes

Syntax

- ASCII [USAGE|TABLE|<value>]

Provenance

- Topic key: DOT|ASCII
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

AT

AT

- Catalog/topic: FOX / AT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the 1-based position of <cSearch> within <cExpression> (case-sensitive).

Status

- implemented=yes; supported=yes

Syntax

- AT(<cSearch>, <cExpression>[, <nOccur>])

Provenance

- Topic key: FOX|AT
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ATC

ATC

- Catalog/topic: FOX / ATC
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the 1-based position of <cSearch> within <cExpression> (case-insensitive).

Status

- implemented=no; supported=yes

Syntax

- ATC(<cSearch>, <cExpression>[, <nOccur>])

Provenance

- Topic key: FOX|ATC
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

AUTODBF

AUTODBF

- Catalog/topic: DOT / AUTODBF
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Create or infer a DBF table from an input source using DotTalk++ automatic DBF-generation rules.

- Create an X64 DBF from a CSV file and import the CSV rows into the newly created table. CSV headers may become field names; when there is no header, deterministic FIELDnnn names are generated.

Status

- implemented=yes; supported=yes

Syntax

- AUTODBF USAGE
- AUTODBF <table> FROM <csvfile>
- AUTODBF X64 <table> FROM <csvfile>
- AUTODBF <table> FROM <csvfile> HEADER
- AUTODBF <table> FROM <csvfile> NOHEADER
- AUTODBF <table> FROM <csvfile> AUTO
- AUTODBF <table> FROM <csvfile> TEXTONLY
- AUTODBF <table> FROM <csvfile> INFER
- AUTODBF <table> FROM <csvfile> OVERWRITE
- AUTODBF [USAGE|<source> [TO <dbf>]]

Usage

- AUTODBF USAGE
- AUTODBF <table> FROM <csvfile>
- AUTODBF X64 <table> FROM <csvfile>
- AUTODBF <table> FROM <csvfile> HEADER
- AUTODBF <table> FROM <csvfile> NOHEADER
- AUTODBF <table> FROM <csvfile> AUTO
- AUTODBF <table> FROM <csvfile> TEXTONLY
- AUTODBF <table> FROM <csvfile> INFER
- AUTODBF <table> FROM <csvfile> OVERWRITE

Argument

- NOHEADER
- Mined command argument/switch candidate. Promote only after validation against parser behavior or curated command docs.

Note

- AUTODBF defaults to X64, AUTO header detection, INFER types, comma CSV.
- AUTO is conservative: it chooses HEADER only when the first row looks like
- names and later data strongly indicates typed data. Use HEADER or NOHEADER
- to remove ambiguity.
- Field names are normalized to command-safe x64 logical names, uniquified,
- capped to the current x64 logical-name limit, and then passed through the
- existing x64 descriptor fallback/mangling policy.
- Long text is not auto-promoted to M yet; values must fit current fixed-field
- x64base limits. This avoids silently writing memo object id 0.
- Existing target DBFs are not overwritten unless OVERWRITE is supplied.
- risk:
- reads_files: yes
- creates_files: yes
- possible_overwrite: only with OVERWRITE
- closes_current_area: yes after validation succeeds
- opens_area: yes

Provenance

- Topic key: DOT|AUTODBF
- Included HELP rows: 40
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BIBLETALK

BIBLETALK

- Catalog/topic: DOT / BIBLETALK
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Use the BibleTalk educational seed database for quote, lookup, and teaching/demo workflows.

Status

- implemented=yes; supported=yes

Syntax

- BIBLETALK [USAGE|QUOTE|SEARCH <text>|BOOKS]

Provenance

- Topic key: DOT|BIBLETALK
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BOTTOM

BOTTOM

- Catalog/topic: DOT / BOTTOM
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Move to the last record in the current table/order.

- Move the current work-area cursor to the last logical record
- Move the current work-area cursor to the last visible/logical record.

Status

- implemented=yes; supported=yes

Syntax

- BOTTOM
- BOTTOM USAGE

Usage

- BOTTOM
- BOTTOM USAGE

Example

- BOTTOM

Note

- Uses the active order when an order is active
- Equivalent user intent to GO BOTTOM / GO LAST
- BOTTOM with no arguments moves to the last visible/logical record.
- BOTTOM requires an open table except for BOTTOM USAGE.
- BOTTOM uses the AutoByFilter last-record navigation selector.
- BOTTOM mutates cursor position but does not mutate table data.
- TALK ON prints the resulting record number when movement succeeds.
- risk:
- mutates__cursor: yes
- mutates__table__data: no
- requires__open__table: yes except usage

Related

- TOP
- BOTTOM
- FIRST
- LAST
- NEXT
- PRIOR
- SKIP
- GOTO
- GPS

Provenance

- Topic key: DOT|BOTTOM
- Included HELP rows: 29
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BROWSE

BROWSE

- Catalog/topic: DOT / BROWSE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Open the classic browse surface for the current table/work-area context.

- Enter the refactored BROWSE module through the legacy global command symbol, preserving existing callers while delegating implementation.

Status

- implemented=yes; supported=yes

Syntax

- BROWSE USAGE
- BROWSE
- BROWSE EDIT

Usage

- BROWSE USAGE
- BROWSE
- BROWSE EDIT

Note

- BROWSE is a thin forwarder to the browse module.
- BROWSE with no arguments enters interactive browse mode.
- BROWSE EDIT requests edit-capable browse behavior where supported by the module.
- Side effects depend on browse actions and delegated commands.
- risk:
- interactive: yes
- mutates_data: possible through edit actions
- delegates_to__browse_module: yes

Related

- BROWSER
- BROWSETUI
- LIST
- DISPLAY
- REPLACE
- Public entrypoint from the new module
- namespace {
- static std::string browse_trim(std::string s)
- {
- while (!s.empty() && std::isspace(static_cast<unsigned char>(s.front()))) s.erase(s.begin());
- while (!s.empty() && std::isspace(static_cast<unsigned char>(s.back()))) s.pop_back();
- return s;
- }
- static std::string browse_upper(std::string s)
- std::transform(s.begin(), s.end(), s.begin(),
- { return static_cast<char>(std::toupper(c)); });
- static bool is_browse_usage_request(std::string raw)
- std::string t = browse_upper(browse_trim(std::move(raw)));
- if (t.rfind("BROWSE", 0) == 0) t = browse_trim(t.substr(7));
- return t == "USAGE" || t == "HELP" || t == "?";
- static void print_browse_usage()
- std::cout
- « "Usage:n"

Provenance

- Topic key: DOT|BROWSE
- Included HELP rows: 40
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BROWSER

BROWSER

- Catalog/topic: DOT / BROWSER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer browser command (experimental).

- Enter a minimal interactive browser with list, display, goto, edit, help, and quit commands, delegating work to existing command handlers.

Status

- implemented=yes; supported=yes

Syntax

- BROWSER USAGE
- BROWSER
- BROWSER EDIT

Usage

- BROWSER USAGE
- BROWSER
- BROWSER EDIT

Note

- BROWSER with no arguments enters interactive browse mode.
- BROWSER EDIT displays edit mode in the banner; edit command is available inside the browser.
- Inside BROWSER, L lists records, D displays current record, G moves to a record, E edits via REPLACE, H or question-mark shows help, and Q quits.
- BROWSER itself is interactive; side effects come from delegated GOTO and REPLACE actions.
- risk:
- interactive: yes
- mutates_record_pointer: GOTO actions
- mutates_table_data: edit actions through REPLACE
- delegates_to_replace: yes

Related

- BROWSE
- BROWSETUI
- LIST
- DISPLAY
- GOTO
- REPLACE
- — Forward declarations of existing command handlers —————
- void cmd_LIST (xbase::DbArea&, std::istream&);
- void cmd_DISPLAY(xbase::DbArea&, std::istream&);
- void cmd_GOTO (xbase::DbArea&, std::istream&);
- void cmd_REPLACE(xbase::DbArea&, std::istream&);
- — Helpers —————
- static inline std::string trim(const std::string& s) {
- size_t b = 0, e = s.size();
- while (b < e && std::isspace(static_cast<unsigned char>(s[b]))) ++b;
- while (e > b && std::isspace(static_cast<unsigned char>(s[e-1]))) --e;
- return s.substr(b, e - b);
- }
- static inline std::string upper(std::string s) {
- std::transform(s.begin(), s.end(), s.begin(),
- { return static_cast<char>(std::toupper(c)); });
- return s;

Provenance

- Topic key: DOT|BROWSER
- Included HELP rows: 40
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BUFFERING

BUFFERING

- Catalog/topic: ED / BUFFERING
- Status: supported
- Implemented/supported: F / T

- Primary/confidence: EDREF / CATALOG

Summary Buffering means changes are staged before permanent commit. Commands TABLE ON REPLACE ... COMMIT ROLLBACK (planned/deferred in some contexts) Observed model With TABLE ON:

- Buffering means changes are staged before permanent commit.
- Commands
- TABLE ON
- REPLACE ...
- COMMIT
- ROLLBACK (planned/deferred in some contexts)
- Observed model
- With TABLE ON:
- TUPLE may show buffered values immediately
- LIST may still show persisted/indexed values until COMMIT
- Educational point
- Buffering separates:
- working state
- persisted state
- This is a classic database concept and an important teaching tool.

Status

- implemented=no; supported=yes

Syntax

- TABLE BUFFERING

Provenance

- Topic key: ED|BUFFERING
- Included HELP rows: 17
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

BUILDLMDB

BUILDLMDB

- Catalog/topic: DOT / BUILDLMDB
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Build or rebuild the LMDB backing store for the current CDX container; may mutate LMDB/index files but not table records.

- Build or rebuild the LMDB backing store for a CDX container using one LMDB environment per table container and named databases for tags.

Status

- implemented=yes; supported=yes

Syntax

- BUILDLMDB USAGE
- BUILDLMDB
- BUILDLMDB YES
- BUILDLMDB AUTO
- BUILDLMDB NOPROMPT
- BUILDLMDB CLEAN YES
- BUILDLMDB FORCE YES
- BUILDLMDB QUIET
- BUILDLMDB SILENT
- BUILDLMDB TINY
- BUILDLMDB SMALL
- BUILDLMDB MEDIUM
- BUILDLMDB LARGE
- BUILDLMDB XL
- BUILDLMDB HUGE
- BUILDLMDB MAPSIZE <size> YES
- BUILDLMDB [HELP|?] [MAPSIZE <n[K|M|G]>|SIZE <n[K|M|G]>|TINY|SMALL|MEDIUM|LARGE|XL|HUGE]
[YES|AUTO|NOPROMPT] [CLEAN|FORCE] [QUIET]

Usage

- BUILDLMDB USAGE
- BUILDLMDB
- BUILDLMDB YES
- BUILDLMDB AUTO
- BUILDLMDB NOPROMPT
- BUILDLMDB CLEAN YES
- BUILDLMDB FORCE YES
- BUILDLMDB QUIET
- BUILDLMDB SILENT
- BUILDLMDB TINY
- BUILDLMDB SMALL
- BUILDLMDB MEDIUM
- BUILDLMDB LARGE
- BUILDLMDB XL
- BUILDLMDB HUGE
- BUILDLMDB MAPSIZE <size> YES
- BUILDLMDB CLEAN MAPSIZE <size> YES

Argument

- NOPROMPT
- Mined command argument/switch candidate. Promote only after validation against parser behavior or curated command docs.

Note

- BUILDLMDB requires an open table except for usage/help requests.
- The public CDX container resolves under INDEXES and the LMDB backend environment resolves under LMDB.
- If an existing LMDB environment would be destructively rebuilt, explicit YES, AUTO, NOPROMPT, QUIET, or SILENT is required.
- CLEAN and FORCE archive the existing environment before rebuild.
- BUILDLMDB releases active index/order state before destructive rebuild.
- BUILDLMDB rebuilds tag databases from current table data.

Provenance

- Topic key: DOT|BUILDLMDB
- Included HELP rows: 45
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CANARY

CANARY

Status notice: **PARTIAL**; supported=F. Treat this page as limited or review-required evidence.

- Catalog/topic: DOT / CANARY
- Status: **partial**
- Implemented/supported: T / F
- Primary/confidence: DOTREF / CATALOG

Summary CANARY is a registered DotTalk++ command; curated DOTREF support status and help summary are pending.

Status

- implemented=yes; supported=no

Provenance

- Topic key: DOT|CANARY
- Included HELP rows: 2
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CDX

CDX

- Catalog/topic: DOT / CDX
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect or manage CDX container metadata and tag directories.

Status

- implemented=yes; supported=yes

Syntax

- CDX USAGE
- CDX INFO [<path.cdx>]
- CDX TAGS [<path.cdx>]
- CDX CREATE [<path.cdx>]
- CDX ADDTAG <name> [<path.cdx>]
- CDX DROPTAG <name> [<path.cdx>]
- CDX [INFO|TAGS|CREATE|ADDTAG|DROPTAG] [<path.cdx>]

Usage

- CDX USAGE
- CDX INFO [<path.cdx>]
- CDX TAGS [<path.cdx>]
- CDX CREATE [<path.cdx>]
- CDX ADDTAG <name> [<path.cdx>]
- CDX DROPTAG <name> [<path.cdx>]

Note

- CDX with no arguments shows usage and does not default to INFO.
- If no path is supplied, CDX first uses the active CDX path from order state when available.
- Otherwise CDX derives <current_dbf_basename>.cdx through the INDEXES path slot.
- CREATE refuses to overwrite an existing file.
- INFO and TAGS are read-only inspection operations and require an existing file.
- ADDTAG and DROPTAG mutate the CDX container tag directory and require an existing file.
- CDX manages container header/tag metadata; backend tag build data persistence is owned elsewhere.
- risk:
- reads_index_file: INFO TAGS ADDTAG DROPTAG
- creates_index_file: CREATE
- overwrites_index_file: no, CREATE refuses existing target
- mutates_index_metadata: ADDTAG DROPTAG
- mutates_table_data: no
- default_path_uses_order_state: yes
- default_path_uses_indexes_slot: yes

Related

- CNX
- INDEX
- SET CDX
- SET ORDER
- REINDEX
- include “xbase.hpp”

Provenance

- Topic key: DOT|CDX
- Included HELP rows: 36
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CHR

CHR

- Catalog/topic: FOX / CHR
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert an integer ASCII code (0-255) to a 1-character string.

Status

- implemented=yes; supported=yes

Syntax

- CHR(<nAsciiCode>)

Provenance

- Topic key: FOX|CHR
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CHRISTMAS

CHRISTMAS

- Catalog/topic: DOT / CHRISTMAS
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display the DotTalk++ Christmas/holiday console splash screen.

Status

- implemented=yes; supported=yes

Syntax

- CHRISTMAS

Provenance

- Topic key: DOT|CHRISTMAS
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CLOSE

CLOSE

- Catalog/topic: DOT / CLOSE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Close the current table, a selected area, or all open work areas.

- Close the current work area, honoring dirty table-buffer prompts, clearing memo/order/table slot state, and clearing affected relation state.

Status

- implemented=yes; supported=yes

Syntax

- CLOSE USAGE
- CLOSE
- CLOSE ALL
- CLOSE [ALL|<area>|<alias>]

Usage

- CLOSE USAGE
- CLOSE
- CLOSE ALL

Note

- CLOSE with no arguments closes the current work area.
- CLOSE ALL currently clears all relations before closing the current area.
- CLOSE prompts or cancels through dirty table-buffer protection when needed.
- CLOSE runs memo sidecar lifecycle hooks before clearing area identity.
- CLOSE clears active order/index state.
- CLOSE resets table buffering state for the slot to off, clean, and fresh.
- CLOSE is a session/area mutation command; it does not directly mutate table records.
- risk:
- closes_area: yes
- clears_order_state: yes
- closes_memo_backend: yes
- resets_table_buffer_state: yes
- clears_relations_for_table: yes
- clears_all_relations: CLOSE ALL
- dirty_prompt_gate: yes
- mutates_table_data: no

Related

- USE
- WORKSPACE
- TABLE
- COMMIT
- REL
- namespace fs = std::filesystem;
- Provided by the interactive shell.
- extern "C" xbase::XBaseEngine* shell_engine(void);
- namespace {
- static int area_index_from_ref(xbase::DbArea& areaRef) {
- xbase::XBaseEngine* eng = nullptr;
- try { eng = shell_engine(); } catch (...) { eng = nullptr; }
- if (!eng) return -1;
- for (int i = 0; i < xbase::MAX_AREA; ++i) {
- try {

Provenance

- Topic key: DOT|CLOSE
- Included HELP rows: 41
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CMDREL

CMDREL

Status notice: **PENDING**; supported=F. Treat this page as limited or review-required evidence.

- Catalog/topic: DOT / CMDREL
- Status: **pending**
- Implemented/supported: F / F
- Primary/confidence: USAGE_CONTRACT / CURATED

Summary Print the recipe for relating HELP COMMANDS to CMD_ARGS.

Usage

- CMDREL

Note

- CMDREL prints commands but does not execute them or alter relation state.

Provenance

- Topic key: DOT|CMDREL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CNX

CNX

- Catalog/topic: DOT / CNX
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Index container command (CNX multi-tag support).

Status

- implemented=yes; supported=yes

Syntax

- CNX USAGE
- CNX INFO [<path.cnx>]
- CNX TAGS [<path.cnx>]
- CNX CREATE [<path.cnx>]
- CNX ADDTAG <name> [<path.cnx>]
- CNX DROPTAG <name> [<path.cnx>]
- CNX WALK <tag> [<path.cnx>]
- CNX TRACE <tag> [<path.cnx>]
- CNX <name>

Usage

- CNX USAGE
- CNX INFO [<path.cnx>]
- CNX TAGS [<path.cnx>]
- CNX CREATE [<path.cnx>]
- CNX ADDTAG <name> [<path.cnx>]
- CNX DROPTAG <name> [<path.cnx>]
- CNX WALK <tag> [<path.cnx>]
- CNX TRACE <tag> [<path.cnx>]

Argument

- NODE
- Mined command argument/switch candidate. Promote only after validation against parser behavior or curated command docs.

Note

- CNX with no arguments shows usage.
- If no path is supplied, CNX first uses the active CNX path from order state when available.
- Otherwise CNX derives <current_dbf_basename>.cnx through the INDEXES path slot.
- CREATE refuses to overwrite an existing file.
- INFO, TAGS, WALK, and TRACE are read-only inspection/diagnostic operations and require an existing file.
- WALK/TRACE use root_page_off from the CNX tag directory and follow plausible child offsets with loop/depth protection.
- ADDTAG and DROPTAG mutate the CNX container tag directory and require an existing file.
- risk:
- reads_index_file: INFO TAGS WALK TRACE ADDTAG DROPTAG
- creates_index_file: CREATE
- overwrites_index_file: no, CREATE refuses existing target
- mutates_index_metadata: ADDTAG DROPTAG
- mutates_table_data: no
- diagnostic_tree_walk: WALK TRACE
- default_path_uses_order_state: yes
- default_path_uses_indexes_slot: yes

Related

- CDX
- INDEX
- SET CNX
- SET ORDER

Provenance

- Topic key: DOT|CNX
- Included HELP rows: 41
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

COBOL

COBOL

- Catalog/topic: DOT / COBOL
- Status: supported

- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display COBOL-oriented educational/demo material for historical data-processing context.

Status

- implemented=yes; supported=yes

Syntax

- COBOL [USAGE|HELP]

Provenance

- Topic key: DOT|COBOL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CODASYL

CODASYL

- Catalog/topic: DOT / CODASYL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display CODASYL/network-database educational/demo material for historical database context.

- Provide a thin CODASYL teaching veneer over already-open DotTalk++ work areas, simulating owner/member set traversal without a second storage engine.

Status

- implemented=yes; supported=yes

Syntax

- CODASYL USAGE
- CODASYL HELP
- CODASYL MODE ON
- CODASYL MODE OFF
- CODASYL LOAD <world>
- CODASYL SETS
- CODASYL SHOW SET <name>
- CODASYL FIND OWNER <set> <value>
- CODASYL FIND OWNER <owner_alias> <value>
- CODASYL GET FIRST
- CODASYL GET FIRST <set>
- CODASYL GET FIRST <member_alias>
- CODASYL GET NEXT
- CODASYL GET NEXT <set>
- CODASYL GET NEXT <member_alias>
- CODASYL WALK
- CODASYL WALK <set>
- CODASYL WALK <member_alias>

- CODASYL [USAGE|HELP]

Usage

- CODASYL USAGE
- CODASYL HELP
- CODASYL MODE ON
- CODASYL MODE OFF
- CODASYL LOAD <world>
- CODASYL SETS
- CODASYL SHOW SET <name>
- CODASYL FIND OWNER <set> <value>
- CODASYL FIND OWNER <owner_alias> <value>
- CODASYL GET FIRST
- CODASYL GET FIRST <set>
- CODASYL GET FIRST <member_alias>
- CODASYL GET NEXT
- CODASYL GET NEXT <set>
- CODASYL GET NEXT <member_alias>
- CODASYL WALK
- CODASYL WALK <set>
- CODASYL WALK <member_alias>
- CODASYL STATUS

Note

- CODASYL with no arguments shows usage.
- This is a teaching adapter and does not create physical CODASYL storage.
- It uses already-open work areas and named set definitions.
- LOAD installs a predefined set map for a named lesson world.
- FIND OWNER captures the current owner and builds a member snapshot.
- GET FIRST and GET NEXT move through the simulated member ring.
- WALK prints a simulated owner/member ring and preserves the member-area cursor best-effort.
- STATUS reports CODASYL teaching state.

Provenance

- Topic key: DOT|CODASYL
- Included HELP rows: 49
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

COMMANDSHELP

COMMANDSHELP

- Catalog/topic: DOT / COMMANDSHELP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Command help (alias of CMDHELP).

Status

- implemented=yes; supported=yes

Syntax

- COMMANDSHELP

Provenance

- Topic key: DOT|COMMANDSHELP
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

COMMIT

COMMIT

- Catalog/topic: DOT / COMMIT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Commit buffered TABLE updates to disk. Notes (current shakedown observations): - Clears TABLE stale state on success. - May currently trigger full INX rebuild work as

- Apply staged table-buffered changes
- Apply buffered TABLE changes to the current area or all open buffered areas, locking records at commit time and reporting persistence-stage failures.
- Commit buffered TABLE updates to disk.
- Notes (current shakedown observations):
 - Clears TABLE stale state on success.
 - May currently trigger full INX rebuild work as part of the commit path (performance issue).

Status

- implemented=yes; supported=yes

Syntax

- COMMIT
- COMMIT ALL
- COMMIT USAGE
- COMMIT MANUAL
- COMMIT INTERACTIVE
- COMMIT AUTO
- COMMIT ALL MANUAL
- COMMIT ALL INTERACTIVE
- COMMIT ALL AUTO

Usage

- COMMIT USAGE
- COMMIT
- COMMIT ALL
- COMMIT MANUAL
- COMMIT INTERACTIVE
- COMMIT AUTO
- COMMIT ALL MANUAL

- COMMIT ALL INTERACTIVE
- COMMIT ALL AUTO

Example

- COMMIT
- COMMIT ALL

Note

- Applies buffered table changes and clears stale state on success
- Index maintenance should flow through the index subsystem rather than direct backend parsing
- COMMIT with no arguments applies buffered changes for the current area.
- COMMIT ALL applies buffered changes for all open buffered areas.
- TABLE ON buffers changes; COMMIT applies them with record locking.
- MANUAL, INTERACTIVE, and AUTO are accepted for compatibility.
- COMMIT does not rebuild CDX or LMDB containers.
- Legacy INX/IDX and CNX rebuild behavior remains only for legacy index families.
- COMMIT is a data mutation command when buffers contain changes.
- COMMIT is a best-effort buffer apply operation, not an atomic transaction.
- risk:
- writes_dbf_records: yes when buffered changes exist
- writes_memo: when buffered memo changes exist
- record_locking: yes at commit time
- clears_table_buffer_changes: on successful commit
- partial_commit_possible: yes
- cdx_lmdb_rebuild: no

Warning

- COMMIT is a mutation boundary; keep help wording conservative until runtime behavior is verified

Related

- TABLE
- REPLACE
- CALCWRITE
- ROLLBACK

Provenance

- Topic key: DOT|COMMIT
- Included HELP rows: 49
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CONCAT

CONCAT

- Catalog/topic: DOT / CONCAT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Concatenate string arguments. (Available as CALC function; also usable as a command where wired.)

- Concatenate one or more expressions into a single printed string
- Concatenate one or more expressions into a single string and print the result.

Status

- implemented=yes; supported=yes

Syntax

- CONCAT <expr1>[, <expr2> ...]
- STRCAT <expr1>[, <expr2> ...]
- CONCAT(<expr1>, <expr2>, ...)
- CONCAT(<c1>[, <c2> ...]) | CONCAT <args...>

Usage

- CONCAT USAGE
- CONCAT <expr1>[, <expr2> ...]
- STRCAT <expr1>[, <expr2> ...]

Example

- CONCAT "hello", " ", "world"
- CONCAT FNAME, " ", LNAME
- STRCAT("A", "B", "C")

Note

- CONCAT is the shell command surface over the same string-function family used by CALC
- When a table is open, bare identifiers can resolve as fields; otherwise they remain literal text
- STRCAT is an alias of CONCAT
- CONCAT accepts between 1 and 32 arguments.

Alias

- STRCAT

Provenance

- Topic key: DOT|CONCAT
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CONTINUE

CONTINUE

- Catalog/topic: DOT / CONTINUE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Continue a running loop/scan block (scripting control flow).

- Continue a previous LOCATE search, or continue with an explicit FOR predicate, following active order when present.

Status

- implemented=yes; supported=yes

Syntax

- CONTINUE

Usage

- CONTINUE
- CONTINUE USAGE
- CONTINUE FOR <expr>

Note

- CONTINUE with no arguments reuses the active LOCATE/CONTINUE predicate.
- CONTINUE FOR <expr> searches forward from the current record using the supplied predicate.
- CONTINUE follows active order when one is present; otherwise it scans physical order.
- CONTINUE requires an open table except for CONTINUE USAGE.
- CONTINUE mutates cursor/search state but does not mutate table data.

Provenance

- Topic key: DOT|CONTINUE
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

COPY

COPY

- Catalog/topic: DOT / COPY
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Copy table or data content into another target using DotTalk++ copy semantics.

- Copy the current DBF, convert the current table to a target DBF flavor, or copy a filesystem file.

Status

- implemented=yes; supported=yes

Syntax

- COPY <source> TO <target>

Usage

- COPY USAGE
- COPY TO <DBFNAME> [WITH SIDECARS] [OVERWRITE]
- COPY TO <DBFNAME> AS <MSDOS|DBASE|FOX26|FOXPRO|VFP|X64> [OVERWRITE]
- COPY TO <DBFNAME> AS X64 VECTOR [OVERWRITE]
- COPY FILE <SRC> TO <DST> [OVERWRITE]

Example

- COPY TO students_copy
- COPY TO students_x64 AS X64 VECTOR OVERWRITE
- COPY TO students_vfp AS VFP
- COPY TO students_backup WITH SIDECARS OVERWRITE

- COPY FILE source.txt TO tmpsource_copy.txt OVERWRITE

Note

- COPY USAGE prints usage and does not require an open table.
- COPY TO requires an open table.
- COPY FILE does not require an open table.
- WITH SIDECARS applies only to binary COPY TO.
- OVERWRITE is required when the destination already exists.

Provenance

- Topic key: DOT|COPY
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CREATE

CREATE

- Catalog/topic: DOT / CREATE
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Create a new table, structure, or local project artifact through the implemented DotTalk++ creation surface.

- Create a DBF table in the configured DBF path slot using the requested xBase/DBF flavor and field specification.

Status

- implemented=yes; supported=yes

Syntax

- CREATE [USAGE|<args...>]

Usage

- CREATE USAGE
- CREATE <name> (<field> <type>[, ...])
- CREATE MSDOS <name> (<field> <type>[, ...])
- CREATE DBASE <name> (<field> <type>[, ...])
- CREATE FOX26 <name> (<field> <type>[, ...])
- CREATE FOXPRO <name> (<field> <type>[, ...])
- CREATE VFP <name> (<field> <type>[, ...])
- CREATE X64 <name> (<field> <type>[, ...])

Example

- CREATE students (sid N(6), lname C(20), fname C(15))
- CREATE X64 teachers (teacher_id I, full_name C(80), bio M)
- CREATE VFP ledger (acct C(12), amount Y, posted D)

Note

- CREATE with no usable table/field specification shows usage and does not create a file.
- Relative table names resolve through the configured DBF path slot.
- CREATE clears active order state and closes the current area before writing the new table.
- After a successful write, CREATE opens the created table in the current area.
- If any field is M, CREATE attempts automatic memo attach after opening the table.
- X64 CREATE applies descriptor fallback/name policy for DBF descriptor safety.
- Long, duplicate, or descriptor-unsafe X64 field names may receive fallback tokens.
- X64 logical/authoritative metadata names are preserved when they fit the current x64 metadata limits.
- CREATE is a filesystem/schema mutation command; do not classify it as a read-only report command.

Provenance

- Topic key: DOT|CREATE
- Included HELP rows: 24
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

CTOD

CTOD

- Catalog/topic: FOX / CTOD
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert a character date to a Date value using current SET DATE/SET CENTURY.

Status

- implemented=yes; supported=yes

Syntax

- CTOD(<cDate>)

Provenance

- Topic key: FOX|CTOD
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DATE

DATE

- Catalog/topic: FOX / DATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the current system date.

Status

- implemented=yes; supported=yes

Syntax

- DATE()

Provenance

- Topic key: FOX|DATE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DECISION

DECISION

- Catalog/topic: ED / DECISION
- Status: **supported**
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Programming Construct 2: Decision FlowDefinition Decision flow chooses between alternatives based on a condition.Current DotTalk++ form IF <expr> ... ELSE ... ENDIF

- Programming Construct 2: Decision Flow
- Definition
- Decision flow chooses between alternatives based on a condition.
- Current DotTalk++ form
- IF <expr>
- ...
- ELSE
- ENDIF
- Concept
- IF asks a yes/no question.
- If true, the IF branch runs.
- Otherwise, the ELSE branch runs, if present.
- Example
- IF GPA > 3.5
- ECHO HONOR STUDENT
- ECHO REGULAR STUDENT
- Educational importance
- This is how a script becomes selective instead of merely linear.
- Typical uses
 - react to a calculation
 - guard a destructive operation

- – separate normal and exceptional paths
- Teaching point
- Decision logic is about controlling execution, not merely testing truth.

Status

- implemented=no; supported=yes

Syntax

- DECISION / IF

Provenance

- Topic key: ED|DECISION
- Included HELP rows: 26
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DELETE

DELETE

- Catalog/topic: DOT / DELETE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Mark the current record deleted using current table semantics.

- Mark the current record or selected records deleted, honoring filters and applying index delete snapshots in direct-write mode.

Status

- implemented=yes; supported=yes

Syntax

- DELETE

Usage

- DELETE USAGE
- DELETE
- DELETE ALL
- DELETE REST
- DELETE NEXT <n>
- DELETE FOR <field> <op> <value>

Note

- DELETE with no arguments deletes the current record.
- DELETE requires an open table except for DELETE USAGE.
- DELETE honors active SET FILTER in ALL, REST, NEXT, and FOR scans.
- DELETE snapshots target recnos before mutating to avoid active-index traversal mutation.
- Direct-write mode captures index keys before delete and applies index delete snapshots after delete.
- Buffered table mode leaves rebuild or final application to COMMIT.

- DELETE marks fields stale best-effort and refreshes current navigation best-effort.
- If index snapshot or apply fails, data delete may still succeed and a rebuild warning is emitted.

Provenance

- Topic key: DOT|DELETE
- Included HELP rows: 18
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DESCEND

DESCEND

- Catalog/topic: DOT / DESCEND
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Set descending sort direction for the active order/tag.

- Set the active order/tag direction to descending for the current work area.

Status

- implemented=yes; supported=yes

Syntax

- DESCEND

Usage

- DESCEND
- DESCEND USAGE

Note

- DESCEND requires an active order except for DESCEND USAGE.
- DESCEND with no arguments mutates order direction to descending.
- DESCEND does not mutate table records or rebuild indexes.

Provenance

- Topic key: DOT|DESCEND
- Included HELP rows: 9
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DIR

DIR

- Catalog/topic: DOT / DIR
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary List directory or file entries through the DotTalk++ shell surface.

- List a directory or show a single file entry through DotTalk++ path resolution.

Status

- implemented=yes; supported=yes

Syntax

- DIR [<mask>|<path>]

Usage

- DIR
- DIR USAGE
- DIR <path>
- DIR <slot>

Note

- DIR with no arguments lists the configured DBF path.
- DIR <path> lists a directory or prints a single file entry.
- Slot-style paths resolve through the common path resolver.
- DIR is read-only and does not mutate table data or filesystem contents.

Provenance

- Topic key: DOT|DIR
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DO

DO

Status notice: **PARTIAL**; supported=F. Treat this page as limited or review-required evidence.

- Catalog/topic: FOX / DO
- Status: partial
- Implemented/supported: F / F
- Primary/confidence: FOXREF / CATALOG

Summary Execute a program. (Use DOTSCRIPT instead.)

Status

- implemented=no; supported=no

Syntax

- DO <program> [WITH <args>]

Provenance

- Topic key: FOX|D0
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DOTHELP

DOTHELP

- Catalog/topic: DOT / DOTHELP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Show project-native DotTalk++ reference entries from the DOTREF catalog.

Status

- implemented=yes; supported=yes

Syntax

- DOTHELP [<term>]

Usage

- DOTHELP
- DOTHELP USAGE
- DOTHELP <term>
- HELP /DOT <term>

Note

- DOTHELP with no arguments lists project-native commands and subsystems.
- DOTHELP <term> prints a matching dotref entry or search matches.
- DOTHELP USAGE prints usage only.
- HELP /DOT <term> is the related HELP-surface access path.
- DOTHELP is read-only.

Provenance

- Topic key: DOT|DOTHELP
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DOTSCRIPT

DOTSCRIPT

- Catalog/topic: DOT / DOTSCRIPT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Run a DotTalk script file (test harness / automation). Example: DOTSCRIPT run_all_sql_tests.dts

- Run a DotTalk++ script file, resolving bare names through script/test search locations, supporting @file notation, TRACE mode, and one-level subscript nesting.
- Run a DotTalk script file (test harness / automation).
- Example:
- DOTSCRIPT run_all_sql_tests.dts

Status

- implemented=yes; supported=yes

Syntax

- DOTSCRIPT <file.dts>

Usage

- DOTSCRIPT USAGE
- DOTSCRIPT <file>
- DOTSCRIPT @<file>
- DOTSCRIPT TRACE
- DOTSCRIPT TRACE ON
- DOTSCRIPT TRACE OFF
- DOTSCRIPT TRACE <file>
- DOTSCRIPT TRACE @<file>
- DOTSCRIPT TRACE ON <file>
- DOTSCRIPT TRACE OFF <file>
- DOTSCRIPT TRACE ON @<file>
- DOTSCRIPT TRACE OFF @<file>
- DOTSCRIPT <file> OUT <transcript-file>
- DOTSCRIPT <file> OUTPUT <transcript-file>
- DOTSCRIPT TRACE <file> OUT <transcript-file>
- DOTSCRIPT <file> OUT <transcript-file> APPEND

Note

- DOTSCRIPT with no arguments shows usage.
- DOTSCRIPT reads an external script file and executes each nonblank, noncomment line through the shell command executor.
- Script comments/blank lines are ignored when they begin with *, //, &&, or ; after trimming.
- Bare script names try the typed name, .dts extension, scripts/, and tests/ candidates.
- @file notation is accepted and unquoted before path resolution.
- TRACE without a file reports the current trace state and usage.
- TRACE ON/OFF changes global DOTSCRIPT trace state.
- TRACE <file> runs a single script with trace enabled without changing global trace state.
- Nesting is limited to main script plus one subscript.
- DOTSCRIPT itself delegates side effects to the commands inside the script; it is not read-only.
- TEST is intentionally not refactored in this patch; TEST may become a later consumer of shell_transcript.

Provenance

- Topic key: DOT|DOTSCRIPT
- Included HELP rows: 34
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DRAWIO

DRAWIO

- Catalog/topic: DOT / DRAWIO
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Generate or manage Draw.io-oriented diagram artifacts for documentation and teaching workflows.

- Launch diagrams.net, or list/open draw.io files from configured diagram paths.

Status

- implemented=yes; supported=yes

Syntax

- DRAWIO [USAGE|<args...>]

Usage

- DRAWIO USAGE
- DRAWIO
- DRAWIO PATHS
- DRAWIO LIST
- DRAWIO LIST SYSTEM
- DRAWIO LIST USER
- DRAWIO LIST ALL
- DRAWIO OPEN
- DRAWIO OPEN <url-or-path>
- DRAWIO OPEN SYSTEM <n|filename>
- DRAWIO OPEN USER <n|filename>
- DRAWIO OPEN ALL <n|filename>

Note

- DRAWIO with no arguments launches the default diagrams.net URL.
- DRAWIO OPEN with no target also launches the default diagrams.net URL.
- DRAWIO LIST defaults to SYSTEM.
- SYSTEM diagrams come from SETPATH SYSTEM_DIAGRAMS / DIAGRAMS.
- USER diagrams come from SETPATH USER_DIAGRAMS.
- DRAWIO does not mutate table data or workspace state.

Provenance

- Topic key: DOT|DRAWIO
- Included HELP rows: 22
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DTOC

DTOC

- Catalog/topic: FOX / DTOC
- Status: **supported**
- Implemented/supported: T / T

- Primary/confidence: FOXREF / CATALOG

Summary Convert a Date value to a character string using current SET DATE/SET CENTURY.

Status

- implemented=yes; supported=yes

Syntax

- DTOC(<dDate>)

Provenance

- Topic key: FOX|DTOC
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

DUMP

DUMP

- Catalog/topic: DOT / DUMP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Debug: print a structured dump of an internal value/state.

- Dump every record from the current work area in legacy pipe-delimited form.

Status

- implemented=yes; supported=yes

Syntax

- DUMP

Usage

- DUMP
- DUMP USAGE

Note

- DUMP requires an open table except for DUMP USAGE.
- DUMP operates only on the current work area.
- DUMP does not resolve paths and does not open files.
- Deleted records are prefixed with an asterisk marker.
- DUMP iterates by record number and reads each record.
- DUMP is read-only for table data but moves the current cursor during output.

Provenance

- Topic key: DOT|DUMP
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EDIT

EDIT

- Catalog/topic: FOX / EDIT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Edit a field/value interactively (where supported).

Status

- implemented=yes; supported=yes

Syntax

- EDIT <field>

Provenance

- Topic key: FOX|EDIT
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EDUCATIONAL__USE

EDUCATIONAL__USE

- Catalog/topic: ED / EDUCATIONAL__USE
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary A practical study order1. Open a table USE STUDENTS STRUCT FIELDS LIST 52. Learn navigation TOP SKIP 1 BOTTOM RECNO3. Learn ordering SET INDEX TO students.cdx

- A practical study order
- 1. Open a table
- USE STUDENTS
- STRUCT
- FIELDS
- LIST 5
- 2. Learn navigation
- TOP
- SKIP 1

- BOTTOM
- RECNO
- 3. Learn ordering
- SET INDEX TO students.cdx
- SET ORDER TO TAG LNAME
- 4. Learn predicates
- LOCATE LNAME = Taylor
- SMARTLIST 5 FOR GPA >= 3.5
- 5. Learn tuple projection
- TUPLE *
- TUPLE LNAME,FNAME
- 6. Learn relations
- SET RELATION TO SID INTO ENROLL
- REL LIST
- 7. Learn relation traversal
- REL ENUM ...
- 8. Learn buffering
- TABLE ON
- REPLACE ...
- COMMIT
- 9. Learn scripting
- IF / ELSE / ENDIF
- LOOP / ENDLOOP
- SCAN / ENDSCAN
- Teaching principle
- Move from single table -> ordered navigation -> predicate logic ->
- projection -> relations -> scripting.

Status

- implemented=no; supported=yes

Syntax

- HOW TO STUDY THIS SYSTEM

Provenance

- Topic key: ED|EDUCATIONAL_USE
- Included HELP rows: 38
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ELSE

ELSE

- Catalog/topic: DOT / ELSE
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Begin the alternate branch of an IF block when the IF condition is false.

- Empty translation-unit shim for ELSE command ownership.

Status

- implemented=yes; supported=yes

Syntax

- ELSE

Usage

- ELSE usage is owned by the IF/ELSE/ENDIF command implementation.
- This file intentionally exports no command handler.

Note

- This file exists only because ELSE has a cmd_*.cpp translation unit in the
- build tree. Do not add a second ELSE implementation here.

Provenance

- Topic key: DOT|ELSE
- Included HELP rows: 8
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ENDIF

ENDIF

- Catalog/topic: DOT / ENDIF
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary End an IF/ELSE conditional block.

- Close an IF control block.

Status

- implemented=yes; supported=yes

Syntax

- ENDIF

Usage

- ENDIF

Note

- Syntax command paired with IF and ELSE. It does not mutate table data by itself.

Provenance

- Topic key: DOT|ENDIF
- Included HELP rows: 6
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ENDLOOP

ENDLOOP

- Catalog/topic: DOT / ENDLOOP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary End a LOOP block (scripting).

- End the active LOOP block and replay buffered commands through the shell executor.

Status

- implemented=yes; supported=yes

Syntax

- ENDLOOP

Usage

- ENDLOOP
- ENDLOOP USAGE

Note

- ENDLOOP with no arguments executes the active LOOP buffer.
- ENDLOOP requires an active LOOP block except for ENDLOOP USAGE.
- ENDLOOP clears active loop state before replay.
- ENDLOOP replays buffered commands through the registered loop executor.
- ENDLOOP reports when no LOOP is active.
- ENDLOOP may indirectly mutate data, session state, or files depending on buffered commands.

Provenance

- Topic key: DOT|ENDLOOP
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ENDSCAN

ENDSCAN

- Catalog/topic: DOT / ENDSCAN
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary End a SCAN block.

- Close a SCAN loop block.

Status

- implemented=yes; supported=yes

Syntax

- ENDSCAN

Usage

- ENDSCAN

Note

- Syntax command paired with SCAN. It does not mutate table data by itself.

Provenance

- Topic key: DOT|ENDSCAN
- Included HELP rows: 6
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ENDUNTIL

ENDUNTIL

- Catalog/topic: DOT / ENDUNTIL
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary End an UNTIL block (scripting).

- Close an UNTIL loop/control block.

Status

- implemented=yes; supported=yes

Syntax

- ENDUNTIL

Usage

- ENDUNTIL

Note

- Syntax command paired with UNTIL where supported. It does not mutate table data by itself.

Provenance

- Topic key: DOT|ENDUNTIL
- Included HELP rows: 6
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ENDWHILE

ENDWHILE

- Catalog/topic: DOT / ENDWHILE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary End a WHILE block (scripting control flow).

- Close a WHILE loop block.

Status

- implemented=yes; supported=yes

Syntax

- ENDWHILE

Usage

- ENDWHILE

Note

- Syntax command paired with WHILE. It does not mutate table data by itself.

Provenance

- Topic key: DOT|ENDWHILE
- Included HELP rows: 6
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ERASE

ERASE

- Catalog/topic: DOT / ERASE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Erase a file or supported target through the DotTalk++ shell surface.

- Physically delete a DBF table file plus known same-stem sidecars across DBF, INDEXES, and LMDB roots.

Status

- implemented=yes; supported=yes

Syntax

- ERASE <target>

Usage

- ERASE USAGE
- ERASE <table> [CONFIRM]
- ERASE TABLE <table> [CONFIRM]

Example

- ERASE TABLE clients
- ERASE TABLE clients CONFIRM
- ERASE students.dbf CONFIRM

Note

- ERASE USAGE prints usage and does not inspect or delete files.
- Without CONFIRM, ERASE performs a dry-run and lists files that would be deleted.
- CONFIRM physically deletes the DBF, matching index containers/files, and matching LMDB backend directory when present.

Provenance

- Topic key: DOT|ERASE
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ERP

ERP

- Catalog/topic: DOT / ERP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display ERP-oriented educational/demo material connecting database concepts to business systems.

- Translation-unit shim for ERP/LabTalk integration includes.

Status

- implemented=yes; supported=yes

Syntax

- ERP [USAGE|HELP]

Usage

- This file currently does not export a command handler.
- If ERP becomes user-facing from this file, add the runtime command handler and full usage contract together.

Note

- Keep ERP application behavior owned by the ERP/LabTalk application layer.

Provenance

- Topic key: DOT|ERP
- Included HELP rows: 7
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ERSATZ

ERSATZ

- Catalog/topic: DOT / ERSATZ
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Launch or drive the Ersatz relational browser/demo surface for workspace and relation-graph inspection.

- Relational browser and tuple-stream helper for current workspace/session, with navigation, tree/grid rendering, workspace handoff, and delta baselines.

Status

- implemented=yes; supported=yes

Syntax

- ERSATZ [USAGE|<args...>]

Usage

- ERSATZ
- ERSATZ USAGE
- ERSATZ SHOW
- ERSATZ REFRESH
- ERSATZ TREE
- ERSATZ GRID
- ERSATZ STATUS
- ERSATZ ORDER
- ERSATZ TOP
- ERSATZ BOTTOM
- ERSATZ NEXT
- ERSATZ NEXT <n>
- ERSATZ PREV
- ERSATZ PREV <n>
- ERSATZ SKIP <n>
- ERSATZ ROOT
- ERSATZ ROOT <alias>
- ERSATZ LIMIT <n>
- ERSATZ PATH <alias>
- ERSATZ CLEARPATH
- ERSATZ BACK
- ERSATZ OPEN <workspace>
- ERSATZ LOAD <name>

- ERSATZ SAVE <name>
- ERSATZ WLOAD <name>
- ERSATZ DELTA MARK <name>
- ERSATZ DELTA SHOW <name>
- ERSATZ DELTA CLEAR <name>
- ERSATZ DELTA CLEAR ALL
- ERSATZ DELTA STATUS
- ERSATZ RESET

Note

- ERSATZ with no arguments renders the current relational browser snapshot.
- SHOW, REFRESH, TREE, and GRID render the current browser session.
- TOP, BOTTOM, NEXT, PREV, and SKIP navigate the root cursor and render.
- ROOT, LIMIT, PATH, CLEARPATH, and BACK mutate browser session settings.
- OPEN hands off to WORKSPACE.
- LOAD, SAVE, and WLOAD read or write workspace files.
- DELTA commands manage in-memory tuple-stream baselines.
- RESET clears ERSATZ browser session state.
- ERSATZ is not table-data mutation by itself, but it can mutate cursor/session/workspace state.

Provenance

- Topic key: DOT|ERSATZ
- Included HELP rows: 44
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EVAL

EVAL

- Catalog/topic: DOT / EVAL
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: DOTREF / CATALOG

Summary Evaluate an expression and print its value.

Status

- implemented=no; supported=yes

Syntax

- EVAL <expr>

Provenance

- Topic key: DOT|EVAL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EVALUATE

EVALUATE

- Catalog/topic: DOT / EVALUATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Evaluate an expression through the DotTalk++ expression layer and display the result.

Status

- implemented=yes; supported=yes

Syntax

- EVALUATE <expr>

Provenance

- Topic key: DOT|EVALUATE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EXAMPLE

EXAMPLE

- Catalog/topic: DOT / EXAMPLE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Show or run small DotTalk++ example/demo material for teaching and verification workflows.

- Minimal example/test command used to verify token parsing and command routing.

Status

- implemented=yes; supported=yes

Syntax

- EXAMPLE [USAGE|<name>|LIST]

Usage

- EXAMPLE USAGE
- EXAMPLE TEST

Note

- EXAMPLE with no arguments shows usage.
- EXAMPLE TEST prints OK.
- EXAMPLE is read-only and does not mutate table data.

Provenance

- Topic key: DOT|EXAMPLE
- Included HELP rows: 9
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EXPFUNCS

EXPFUNCS

- Catalog/topic: DOT / EXPFUNCS
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Export the expression/function catalog to Markdown documentation.

Status

- implemented=yes; supported=yes

Syntax

- EXPFUNCS [MD [<path>]]

Provenance

- Topic key: DOT|EXPFUNCS
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EXPORT

EXPORT

- Catalog/topic: DOT / EXPORT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Export to CSV.

- Export the current DBF rowset, or an already-open named work area, to a delimited file.

Status

- implemented=yes; supported=yes

Syntax

- EXPORT <csv>

Usage

- EXPORT USAGE
- EXPORT [TO] <file> [CSV|PIPE]
- EXPORT <open-area-token> TO <file> [CSV|PIPE]

Note

- EXPORT [TO] <file> writes the current table to the named file.
- EXPORT <open-area-token> TO <file> writes an already-open work area without changing the user's selected area intentionally.
- Named tokens may be an area number, #area, alias/name, logical name, DBF basename/stem, filename, or full path, if those values resolve uniquely to an open area.
- Named EXPORT does not auto-open tables from disk.
- CSV is the default format; PIPE uses a pipe delimiter.
- A missing extension is added automatically (.csv for CSV, .txt for PIPE).
- EXPORT writes a header row.
- EXPORT honors the active SET FILTER for the exported area.
- EXPORT reads records in physical table order.
- EXPORT may report file/write errors and still emit a summary when appropriate.

Provenance

- Topic key: DOT|EXPORT
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EXPORTSQL

EXPORTSQL

- Catalog/topic: DOT / EXPORTSQL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Export DotTalk++ table or workspace data through the SQL export helper surface.

- Preview SQL export output and expose the file-export command surface.

Status

- implemented=yes; supported=yes

Syntax

- EXPORTSQL [USAGE|<args...>]

Usage

- EXPORTSQL USAGE
- EXPORTSQL PREVIEW <table>
- EXPORTSQL FILE <table> TO <file>

Example

- EXPORTSQL PREVIEW students
- EXPORTSQL FILE students TO tmpstudents.sql

Note

- EXPORTSQL USAGE returns before file or table work.
- EXPORTSQL hooks are currently preview/file command surfaces.

Provenance

- Topic key: DOT|EXPORTSQL
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

EXPRESSION

EXPRESSION

- Catalog/topic: ED / EXPRESSION
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary An expression is something the engine evaluates to produce a value. Kinds of expressions numeric character date logical Examples 1 + 2 UPPER("abc") GPA + 0.01 DATE()

- An expression is something the engine evaluates to produce a value.
- Kinds of expressions
 - numeric
 - character
 - date
 - logical
- Examples
 - 1 + 2
 - UPPER("abc")
 - GPA + 0.01
 - DATE() + 2
 - LNAME = "TAYLOR"
- Commands using expressions
 - CALC
 - EVAL
 - REPLACE
 - IF
 - LOCATE
 - SMARTLIST
- Teaching point
- Expressions produce values.
- Predicates are expressions whose value is true or false.

Status

- implemented=no; supported=yes

Syntax

- EXPRESSIONS

Provenance

- Topic key: ED|EXPRESSION
- Included HELP rows: 24
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

FIND

FIND

- Catalog/topic: DOT / FIND
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Find text or values using the active order when possible, with scan fallback when needed.

- Seek an exact value using the active index/order
- Find text in the current table, delegating to SEEK when the active order can satisfy the request and otherwise scanning the selected field.

Status

- implemented=yes; supported=yes

Syntax

- SEEK <expr>
- SEEK <expr> IN <alias>
- FIND <text> [IN <field>]

Usage

- FIND USAGE
- FIND <text>
- FIND <field> <text>
- FIND <text> IN <field>

Example

- SEEK "SMITH"
- SEEK 1001
- SEEK DTOS(DATE())

Note

- Requires an active table and normally an active order/index
- Successful SEEK intentionally moves the current record pointer
- Command wording should stay at the CDX/tag/order level; LMDB is a backend detail
- FIND requires an open table except for FIND USAGE.
- FIND with one text argument delegates to SEEK when an order is active.
- FIND with a field delegates to SEEK only when that field is the active tag.
- Otherwise FIND scans the requested field using ordered or physical traversal.
- FIND honors active SET FILTER visibility.
- FIND positions on the found record when successful.
- FIND restores the prior cursor when not found.

Warning

- Do not document direct LMDB path parsing as public SEEK behavior

Provenance

- Topic key: DOT|FIND
- Included HELP rows: 25
- HELP reference run: MANRUN-20260717T222026Z-28C704E0

- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

FOXHELP

FOXHELP

- Catalog/topic: DOT / FOXHELP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary List or search command help topics. Examples: FOXHELP FOXHELP WORKSPACE FOXHELP REL Notes: Uses the help catalogs and any HELP DBFs.

- List or search the static FoxPro-style command catalog.
- List or search command help topics.
- Examples:
- FOXHELP
- FOXHELP WORKSPACE
- FOXHELP REL
- Notes:
- Uses the help catalogs and any HELP DBFs.

Status

- implemented=yes; supported=yes

Syntax

- FOXHELP [<term>]

Usage

- FOXHELP
- FOXHELP USAGE
- FOXHELP <name>
- FOXHELP <search>
- FH
- FH <name>
- FH <search>

Note

- FOXHELP with no arguments lists the FoxPro-style command subset.
- FOXHELP <name> prints an exact catalog item when found.
- FOXHELP <search> searches the catalog and prints matching items.
- FH is a short alias for FOXHELP.
- FOXHELP is a read-only help/report command.

Provenance

- Topic key: DOT|FOXHELP
- Included HELP rows: 22
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

FOXPRO

FOXPRO

- Catalog/topic: DOT / FOXPRO
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary DotTalk++ UI / browser command.

Status

- implemented=yes; supported=yes

Syntax

- FOXPRO

Provenance

- Topic key: DOT|FOXPRO
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

FOXSTANDARD

FOXSTANDARD

- Catalog/topic: DOT / FOXSTANDARD
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display FoxPro/xBase standard-reference or compatibility teaching material.

- Render static historical FoxPro-standard reference topics.

Status

- implemented=yes; supported=yes

Syntax

- FOXSTANDARD [USAGE|HELP|<topic>]

Usage

- FOXSTANDARD USAGE
- FOXSTANDARD <command>
- FOXSTANDARD ALL
- FOXSTANDARD TOPICS
- FOXSTANDARD LIST

Note

- FOXSTANDARD with no arguments shows usage.
- FOXSTANDARD ALL, TOPICS, and LIST render the available topic list.
- FOXSTANDARD <command> renders the static reference for that command.
- FOXSTANDARD is separate from the live HELP and command catalogs.

Provenance

- Topic key: DOT|FOXSTANDARD
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

FOXTALK

FOXTALK

- Catalog/topic: DOT / FOXTALK
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Legacy alias for the ArcticTalk Turbo Vision TUI shell. Example: FOXTALK Notes: Intended for keyboard-driven browsing and diagnostics. Exits back

- Legacy alias for the ArcticTalk Turbo Vision TUI shell.
- Example:
- FOXTALK
- Notes:
- Intended for keyboard-driven browsing and diagnostics.
- Exits back to the DotTalk++ CLI.

Status

- implemented=yes; supported=yes

Syntax

- FOXTALK

Provenance

- Topic key: DOT|FOXTALK
- Included HELP rows: 8
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

GLOSSARY

GLOSSARY

- Catalog/topic: ED / GLOSSARY
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Quick glossaryAlias Name used to refer to a table/work area.Buffer Temporary staged change area before persistence.Child Related table reached from a parent.Current area The ac

- Quick glossary
- Alias
- Name used to refer to a table/work area.
- Buffer
- Temporary staged change area before persistence.
- Child
- Related table reached from a parent.
- Current area
- The active work area.
- Current record
- The row at the current record pointer.
- Enum / enumerate
- Emit results one by one.
- Field
- A named value within a record.
- Index
- Ordered access path.
- Metadata
- Structural information about data.
- Order
- Current sequence used for navigation.
- Parent
- Table from which relation traversal begins.
- Predicate
- True/false expression.
- Projection
- Selection of fields/columns for output.
- Record
- One row.
- Relation
- Parent-child matching rule between tables.
- Schema
- Structural definition of a table/dataset.
- Sequential flow
- Execute one command after another.
- State
- The live session condition.
- Tuple
- Logical projected row.
- Work area
- Active open-table slot.

Status

- implemented=no; supported=yes

Syntax

- GLOSSARY

Provenance

- Topic key: ED|GLOSSARY
- Included HELP rows: 43
- HELP reference run: MANRUN-20260717T222026Z-28C704E0

- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

GO

GO

- Catalog/topic: DOT / GO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Execute a DotTalk++ command line (dispatch to command handlers).

- Move to a record or navigation endpoint
- Refresh the current record, move to table/order endpoints, move to an absolute record number, or skip relative to the current record.

Status

- implemented=yes; supported=yes

Syntax

- GO
- GO TOP
- GO BOTTOM
- GO FIRST
- GO LAST
- GO [TO] <recno>
- GO RECORD <recno>
- GO +<n>
- GO -<n>
- GO <commandLine>

Usage

- GO
- GO USAGE
- GO TOP
- GO BOTTOM
- GO FIRST
- GO LAST
- GO TO <recno>
- GO RECORD <recno>
- GO <recno>
- GO +<n>
- GO -<n>

Example

- GO
- GO TOP
- GO TO 10
- GO RECORD 25
- GO +5
- GO -1

Note

- GO with no arguments refreshes or reports the current record context
- TOP and FIRST are synonyms
- BOTTOM and LAST are synonyms
- Relative movement uses the current work-area cursor
- GO with no arguments refreshes/re-reads the current record through the navigation layer.
- GO TOP/BOTTOM/FIRST/LAST move to logical endpoints.
- GO <recno>, GO TO <recno>, and GO RECORD <recno> navigate absolutely.
- GO +/-<n> delegates to relative skip.
- GO USAGE prints usage before navigation.

Warning

- IN <alias> is not supported yet

Provenance

- Topic key: DOT|GO
- Included HELP rows: 41
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

GOTO

GOTO

- Catalog/topic: DOT / GOTO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Move the current work area to a specific record or boundary position.

- Move to a record or navigation endpoint
- Move the current work-area cursor to an absolute record number, first record, or last record through the shared navigation layer.

Status

- implemented=yes; supported=yes

Syntax

- GO
- GO TOP
- GO BOTTOM
- GO FIRST
- GO LAST
- GO [TO] <recno>
- GO RECORD <recno>
- GO +<n>
- GO -<n>
- GOTO <recno>|TOP|BOTTOM

Usage

- GOTO USAGE
- GOTO <recno>

- GOTO FIRST
- GOTO LAST

Example

- GO
- GO TOP
- GO TO 10
- GO RECORD 25
- GO +5
- GO -1

Note

- GO with no arguments refreshes or reports the current record context
- TOP and FIRST are synonyms
- BOTTOM and LAST are synonyms
- Relative movement uses the current work-area cursor
- GOTO requires a target argument except for GOTO USAGE.
- GOTO FIRST and GOTO LAST use endpoint navigation.
- GOTO <recno> uses absolute record navigation.
- GOTO mutates cursor position but does not mutate table data.

Warning

- IN <alias> is not supported yet

Provenance

- Topic key: DOT|GOTO
- Included HELP rows: 33
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

GPS

GPS

- Catalog/topic: DOT / GPS
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Report current session/navigation position and work-area orientation diagnostics.

- Report current work-area position, including area slot, table label, physical record number, and computed logical row.

Status

- implemented=yes; supported=yes

Syntax

- GPS

Usage

- GPS
- GPS USAGE

Note

- GPS with no arguments reports cursor position.
- GPS with no open table reports the current area and no-table state.
- GPS computes logical row by iterating visible ordered records.
- GPS is read-only for table data but may temporarily move the cursor while computing logical row.

Provenance

- Topic key: DOT|GPS
- Included HELP rows: 10
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

IDX

IDX

- Catalog/topic: DOT / IDX
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Memory-only educational index lab for teaching sort and index concepts without writing persistent index files.

- Memory-only educational index lab for teaching sorting and index concepts without writing persistent INX/CNX/CDX files.

Status

- implemented=yes; supported=yes

Syntax

- `IDX [USAGE|LIST|DROP <tag>|DROP ALL|ON <field|#n> TAG <name> [SORT <algo>|<algo>] [ASC|DESC]]`

Usage

- IDX
- IDX USAGE
- IDX ON <field|#n> TAG <name>
- IDX ON <field|#n> TAG <name> SORT <algo>
- IDX ON <field|#n> TAG <name> <algo>
- IDX ON <field|#n> TAG <name> ASC
- IDX ON <field|#n> TAG <name> DESC
- IDX LIST
- IDX DROP <tag>
- IDX DROP ALL

Example

- `IDX ON LNAME TAG lname_std`
- `IDX ON LNAME TAG lname_bubble BUBBLE`
- `IDX ON LNAME TAG lname_bubble2 SORT BUBBLE DESC`

Note

- `IDX` with no arguments prints help/usage.
- `IDX` is memory-only and does not write `.inx` files.
- `IDX` does not participate in `SET ORDER`, `REINDEX`, `WORKSPACE restore`, `IndexManager`, or `IIndexBackend`.
- Use `INDEX` for persistent index files.
- `SORT` algorithms currently include `STD` and `BUBBLE`.

Provenance

- Topic key: `DOT|IDX`
- Included `HELP` rows: 22
- `HELP` reference run: `MANRUN-20260717T222026Z-28C704E0`
- Disposition run: `MANRUN-20260717T230554Z-DB3F2DC8`
- Authority: `candidate_only; publication_authority_claimed=0`

IF

IF

- Catalog/topic: `DOT / IF`
- Status: `supported`
- Implemented/supported: `T / T`
- Primary/confidence: `DOTREF / CATALOG`

Summary Begin a conditional DotScript block; execute the following block when the logical expression is true.

- Start an `IF/ELSE/ENDIF` conditional block using the shell's shared boolean expression evaluator and control-flow stack.

Status

- `implemented=yes; supported=yes`

Syntax

- `IF <logical_expr>`

Usage

- `IF USAGE`
- `IF <bool-expr>`
- `ELSE`
- `ELSE USAGE`
- `ENDIF`
- `ENDIF USAGE`

Example

- `IF GPA >= 3.0`
- `ECHO HONORS`
- `ELSE`
- `ECHO REGULAR`
- `ENDIF`

Note

- IF USAGE prints usage and does not modify the IF stack.
- IF evaluates only when the outer IF stack allows execution.
- ELSE flips the active branch for the current IF frame.
- ENDIF exits the current IF frame.
- Effects of commands inside the active branch are owned by those commands.

Provenance

- Topic key: DOT|IF
- Included HELP rows: 20
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

IMAGE

IMAGE

- Catalog/topic: DOT / IMAGE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect image file metadata or open a supported image file in the operating-system viewer.

- Inspect image file metadata or open a supported image file in the operating system viewer.

Status

- implemented=yes; supported=yes

Syntax

- IMAGE [USAGE|INFO <file>|<file>]

Usage

- IMAGE USAGE
- IMAGE <file>
- IMAGE INFO <file>

Note

- IMAGE with no arguments prints usage.
- IMAGE USAGE prints usage and does not open a viewer.
- IMAGE INFO <file> prints file extension, size, and recognized-image status.
- IMAGE <file> opens the OS viewer on Windows.
- Non-Windows viewer launch is currently not implemented.
- IMAGE does not mutate table data.

Provenance

- Topic key: DOT|IMAGE
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

IMPORT

IMPORT

- Catalog/topic: DOT / IMPORT
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Import from CSV.

- Import records from a CSV file into the current open table by matching CSV headers to field names case-insensitively.

Status

- implemented=yes; supported=yes

Syntax

- IMPORT <csv>

Usage

- IMPORT USAGE
- IMPORT <csvfile>

Note

- IMPORT requires an open table except for IMPORT USAGE.
- IMPORT appends .csv to the file name when the extension is omitted.
- The first CSV row is interpreted as headers.
- Headers are mapped to current table fields case-insensitively.
- Each data row appends a blank record, sets mapped fields, and writes the record.
- Unmapped CSV columns are ignored.
- IMPORT mutates table data by appending records.

Provenance

- Topic key: DOT|IMPORT
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

IMPORTSQL

IMPORTSQL

- Catalog/topic: DOT / IMPORTSQL
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Import or bridge SQL data into DotTalk++ through the SQL import helper surface.

- Preview, validate, infer schema, create/import table data from delimited files.

Status

- implemented=yes; supported=yes

Syntax

- IMPORTSQL [USAGE|<args...>]

Usage

- IMPORTSQL USAGE
- IMPORTSQL PREVIEW <file> [DELIM PIPE|TAB|COMMA]
- IMPORTSQL VALIDATE <file> [DELIM PIPE|TAB|COMMA]
- IMPORTSQL SCHEMA <file> [DELIM PIPE|TAB|COMMA]
- IMPORTSQL CREATE <file> TO <table> [DELIM PIPE|TAB|COMMA]
- IMPORTSQL FILE <file> TO <table> [DELIM PIPE|TAB|COMMA]
- IMPORTSQL MAP <subcommand> <mapfile>

Example

- IMPORTSQL PREVIEW datastudents.psv
- IMPORTSQL VALIDATE datastudents.csv DELIM COMMA
- IMPORTSQL CREATE datastudents.psv TO students
- IMPORTSQL FILE datastudents.psv TO students

Note

- IMPORTSQL USAGE returns before file/table work.
- IMPORTSQL PREVIEW/VALIDATE/SCHEMA read input files.
- IMPORTSQL CREATE/FILE may create tables and import records.

Provenance

- Topic key: DOT|IMPORTSQL
- Included HELP rows: 18
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

INDEX

INDEX

- Catalog/topic: DOT / INDEX
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary General index-management command surface for the current table.

- Build an INX index file from the current table using a field key, tag/file name, optional direction, and optional INX format.

Status

- implemented=yes; supported=yes

Syntax

- INDEX [ON <field> TAG <name> | STATUS | LIST]

Usage

- INDEX USAGE
- INDEX ON <field> TAG <name>
- INDEX ON <field> TAG <name> ASC
- INDEX ON <field> TAG <name> DESC
- INDEX ON <field> TAG <name> 1INX
- INDEX ON <field> TAG <name> 2INX
- INDEX ON <field> TAG <name> ASC 1INX
- INDEX ON <field> TAG <name> DESC 2INX

Example

- INDEX ON LNAME TAG students
- INDEX ON LNAME TAG students DESC
- INDEX ON LNAME TAG students ASC 1INX
- INDEX ON LNAME TAG students DESC 2INX

Note

- INDEX requires an open table except for INDEX USAGE, INDEX HELP, and INDEX question-mark.
- Deleted records are excluded.
- Default direction is ASC.
- Default output format is 2INX, matching REINDEX.
- Optional direction and format tokens may appear in either order.
- Field-number tokens are also accepted by the parser, but omitted from mineable usage rows because hash syntax is a source-comment marker.
- 2INX uses fixed-length keys, uppercases character fields, and writes a pos-by-recno table.
- TAG must name an INX file target; non-.inx extensions are refused.
- INDEX writes an index file through the INDEXES path resolver and does not mutate table records.

Provenance

- Topic key: DOT|INDEX
- Included HELP rows: 25
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

INIT

INIT

- Catalog/topic: DOT / INIT
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Initialize default paths, perform best-effort stale-lock cleanup, and process startup ini scripts.

- Initialize runtime paths, cleanup stale locks, and run system/user init scripts from the executable directory.

Status

- implemented=yes; supported=yes

Syntax

- INIT USAGE

Usage

- INIT
- INIT USAGE

Note

- INIT with no arguments initializes default paths when needed and reports path slots.
- INIT cleans stale DBF locks best-effort.
- INIT runs dottalkpp.ini and init.ini from the executable directory when present.
- INIT USAGE prints usage and does not initialize paths, cleanup locks, or run scripts.
- Script commands run through the shell command executor and may have their own side effects.

Provenance

- Topic key: DOT|INIT
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

INTRO

INTRO

- Catalog/topic: ED / INTRO
- Status: **supported**
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary DotTalk++ Educational ReferencePurpose EDREF is the instructional help catalog for learning DotTalk++ as a system.What belongs here - programming concepts shown by DotScript and command fl

- DotTalk++ Educational Reference
- Purpose
- EDREF is the instructional help catalog for learning DotTalk++ as a system.
- What belongs here
 - – programming concepts shown by DotScript and command flow
 - – data concepts such as tables, records, fields, indexes, and relations
 - – engine concepts such as metadata, tuple projection, ordering, buffering
 - – practical examples written in the style of DotTalk++ commands
- What does not belong here
 - – FoxPro compatibility syntax reference itself
 - – raw command catalog material already kept in foxref.hpp / dotref.hpp
- Teaching approach
- This file explains the system in plain language.
- It is meant to answer:
 - “What is this feature for?”
 - “How do I think about it?”
 - “What is the simplest example?”

- “How does it relate to the larger engine?”

Status

- implemented=no; supported=yes

Syntax

- INTRO

Provenance

- Topic key: ED|INTRO
- Included HELP rows: 20
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LEFT

LEFT

- Catalog/topic: FOX / LEFT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the left-most <nChars> characters of <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- LEFT(<cExpression>, <nChars>)

Provenance

- Topic key: FOX|LEFT
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LEN

LEN

- Catalog/topic: FOX / LEN
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the length (character count) of <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- LEN(<cExpression>)

Provenance

- Topic key: FOX|LEN
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LIST

LIST

- Catalog/topic: DOT / LIST
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary List records from the current table using DotTalk++ command semantics.

- List records from the current work area
- Display rows from the current work area using classic LIST-style output, honoring active order/index state and optional filtering modes.

Status

- implemented=yes; supported=yes

Syntax

- LIST
- LIST ALL
- LIST DELETED
- LIST FIELDS <fieldlist>
- LIST FOR <expr>
- LIST WHILE <expr>
- LIST NEXT <n>
- LIST REST
- LIST [ALL] [FIELDS <list>] [FOR <expr>]

Usage

- LIST
- LIST USAGE
- LIST TOP
- LIST BOTTOM
- LIST ALL
- LIST <limit>
- LIST DELETED
- LIST FOR <predicate>
- LIST TOP <limit>
- LIST BOTTOM <limit>

Example

- LIST
- LIST FIELDS LNAME,FNAME

- LIST FOR GPA >= 3.5
- LIST NEXT 10

Note

- Developer inspection command; SMARTLIST is the preferred order-aware listing path
- Should preserve the current cursor unless a handler explicitly documents movement
- Filtering is expression/predicate-aware where wired
- LIST requires an open table except for LIST USAGE.
- LIST with no arguments displays from the current cursor position.
- LIST ALL starts at the top and removes the default output limit.
- TOP and BOTTOM move to an endpoint before listing.
- DELETED selects deleted records using physical traversal.
- FOR applies an expression predicate after normalization.
- Active order/index state is honored when possible; LIST falls back to physical order with a note.
- LIST is read-only for table data and restores cursor position best-effort.

Provenance

- Topic key: DOT|LIST
- Included HELP rows: 38
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LMDB

LMDB

- Catalog/topic: DOT / LMDB
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect or manage per-area LMDB-backed index/storage wiring where supported.

- Inspect and control the per-area LMDB backed CDX index backend through the current DbArea IndexManager.

Status

- implemented=yes; supported=yes

Syntax

- LMDB [USAGE|INFO|OPEN|USE|SEEK|DUMP|SCAN|CLOSE] ...

Usage

- LMDB USAGE
- LMDB INFO
- LMDB OPEN <container.cdx>
- LMDB OPEN <envdir.cdx.d>
- LMDB OPEN <stem>
- LMDB USE <tag>
- LMDB SEEK <key>
- LMDB DUMP
- LMDB DUMP <max>
- LMDB SCAN <low> <high>
- LMDB CLOSE

Note

- LMDB is a thin wrapper over the current area IndexManager and CDX backend.
- LMDB does not use LMDB_UTIL or any shared global LMDB environment.
- Bare stems are resolved through the INDEXES path slot.
- OPEN attaches the CDX container and updates legacy order state.
- USE selects an active tag and updates legacy active-tag state.
- SEEK searches the selected tag and reports the matching record number.
- DUMP and SCAN inspect the selected tag.
- CLOSE closes the current area index manager and clears order state.
- LMDB mutates index/order session state but not table records.

Provenance

- Topic key: DOT|LMDB
- Included HELP rows: 24
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LMDB_UTIL

LMDB_UTIL

- Catalog/topic: DOT / LMDB_UTIL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic LMDB utility surface for low-level backend inspection and maintenance workflows.

- Deprecated disabled LMDB utility command that points users to the per-area LMDB command.

Status

- implemented=yes; supported=yes

Syntax

- LMDB_UTIL [USAGE|<args...>]

Usage

- LMDB_UTIL
- LMDB_UTIL USAGE

Note

- LMDB_UTIL is deprecated and disabled.
- LMDB_UTIL intentionally does not open LMDB environments or transactions.
- Use LMDB INFO, LMDB OPEN, LMDB USE, LMDB SEEK, LMDB DUMP, LMDB SCAN, and LMDB CLOSE instead.
- This avoids cross-area contamination and reader-slot conflicts.

Provenance

- Topic key: DOT|LMDB_UTIL
- Included HELP rows: 10
- HELP reference run: MANRUN-20260717T222026Z-28C704E0

- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LMDBDUMP

LMDBDUMP

- Catalog/topic: DOT / LMDBDUMP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic surface for dumping or inspecting LMDB-backed index/storage state.

- Open an LMDB environment read-only and dump keys and values for diagnostics, with optional named DB, grep, limit, and start-key controls.

Status

- implemented=yes; supported=yes

Syntax

- LMDBDUMP [USAGE|<args...>]

Usage

- LMDBDUMP USAGE
- LMDBDUMP <env_path>
- LMDBDUMP <env_path> -db <name>
- LMDBDUMP <env_path> -db <name>
- LMDBDUMP <env_path> -grep <ascii>
- LMDBDUMP <env_path> -grep <ascii>
- LMDBDUMP <env_path> -trydb
- LMDBDUMP <env_path> -limit <n>
- LMDBDUMP <env_path> -start <key>
- LMDBDUMP <env_path> -starthex <hex>

Example

- LMDBDUMP indexesstudents.cdx.d
- LMDBDUMP indexesstudents.cdx.d -trydb
- LMDBDUMP indexesstudents.cdx.d -grep MILLER -limit 50
- LMDBDUMP indexesstudents.cdx.d -db lname -start M -limit 200

Note

- LMDBDUMP opens the supplied LMDB environment read-only.
- LMDBDUMP does not depend on the xindex backend or current work area.
- -start treats the key as ASCII unless it begins with 0x.
- -starthex accepts hex bytes.
- -trydb scans main DB keys and probes named DB candidates.
- LMDBDUMP is diagnostic and does not mutate table or index data.

Provenance

- Topic key: DOT|LMDBDUMP
- Included HELP rows: 24
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8

- Authority: candidate_only; publication_authority_claimed=0

LOCATE

LOCATE

- Catalog/topic: DOT / LOCATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Position on the first record matching a predicate or simple field comparison.

- Locate the first record matching a predicate, using simple CDX fast-path when possible and selector-backed scanning otherwise.

Status

- implemented=yes; supported=yes

Syntax

- LOCATE FOR <expr> | LOCATE <field> <op> <value>

Usage

- LOCATE USAGE
- LOCATE FOR <expr>
- LOCATE <field> <op> <value>

Example

- LOCATE FOR LNAME = Smith
- LOCATE LNAME = Smith
- LOCATE FOR BALANCE > 100

Note

- LOCATE requires an open table except for LOCATE USAGE.
- LOCATE with no predicate shows usage.
- LOCATE clears previous LOCATE and CONTINUE bridge state before searching.
- Simple predicates on the active CDX tag may use the CDX fast path.
- Complex predicates are evaluated through the selector and expression path.
- LOCATE positions on the first matching record.
- LOCATE updates locate state and CONTINUE bridge state after a match.
- LOCATE is read-only for table data but mutates cursor/search state.

Provenance

- Topic key: DOT|LOCATE
- Included HELP rows: 18
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LOCK

LOCK

- Catalog/topic: DOT / LOCK
- Status: supported

- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Lock the current record (or table) for editing.

- Acquire record or table locks for the current table and inspect lock status or lock ownership.

Status

- implemented=yes; supported=yes

Syntax

- LOCK [RECORD|TABLE|<recno>...]

Usage

- LOCK USAGE
- LOCK
- LOCK <n>
- LOCK ALL
- LOCK TABLE
- LOCK STATUS
- LOCK WHO <n>

Note

- LOCK requires an open table except for LOCK USAGE.
- LOCK with no arguments locks the current record.
- LOCK <n> locks record n.
- LOCK ALL and LOCK TABLE lock the entire table.
- LOCK STATUS reports table and current-record lock state.
- LOCK WHO <n> reports the owner of record n when a lock is recorded.
- LOCK mutates lock state but does not mutate table data.

Provenance

- Topic key: DOT|LOCK
- Included HELP rows: 18
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LOOP

LOOP

- Catalog/topic: DOT / LOOP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Begin a LOOP block (scripting).

- Start buffering commands for later replay by ENDLOOP, with optional quiet mode and numeric repetition labels.

Status

- implemented=yes; supported=yes

Syntax

- LOOP FOR <n> TIMES

Usage

- LOOP
- LOOP USAGE
- LOOP QUIET
- LOOP <n>
- LOOP <n> TIMES
- LOOP FOR <n>
- LOOP FOR <n> TIMES
- LOOP FOR <label>
- LOOP OVERRIDE <label>

Note

- LOOP with no arguments starts command buffering and replays once at ENDLOOP.
- LOOP <n>, LOOP <n> TIMES, and LOOP FOR <n> replay buffered commands n times.
- LOOP QUIET suppresses buffering and ENDLOOP status messages.
- LOOP FOR <label> stores a nonnumeric label and currently replays once.
- ENDLOOP executes buffered commands through the pluggable shell executor.
- The loop implementation skips buffered ENDLOOP lines during replay.
- Iteration count is clamped to the hard maximum when necessary.
- LOOP mutates script execution state and may indirectly mutate anything its buffered commands mutate.

Provenance

- Topic key: DOT|LOOP
- Included HELP rows: 21
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LOOP_BUFFER

LOOP_BUFFER

- Catalog/topic: DOT / LOOP_BUFFER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic helper surface for inspecting or testing buffered LOOP control-flow behavior.

Status

- implemented=yes; supported=yes

Syntax

- LOOP_BUFFER [USAGE|<args...>]

Provenance

- Topic key: DOT|LOOP_BUFFER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8

- Authority: candidate_only; publication_authority_claimed=0

LOOPS

LOOPS

- Catalog/topic: ED / LOOPS
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Programming Construct 3: Iteration / Looping Definition A loop repeats work. DotTalk++ loop families LOOP ... ENDLOOP WHILE ... ENDWHILE UNTIL ... ENDUNTIL SCAN ... ENDSCAN1. L

- Programming Construct 3: Iteration / Looping
- Definition
- A loop repeats work.
- DotTalk++ loop families
- LOOP ... ENDLOOP
- WHILE ... ENDWHILE
- UNTIL ... ENDUNTIL
- SCAN ... ENDSCAN
- 1. LOOP
- Repeats a fixed number of times.
- Example
- LOOP 3 TIMES
- ECHO HELLO
- ENDLOOP
- 2. WHILE
- Repeats while a condition stays true.
- Example concept
- WHILE counter < 10
- ...
- ENDWHILE
- 3. UNTIL
- Repeats until a condition becomes true.
- 4. SCAN
- Record-oriented loop over table rows.
- SCAN
- TUPLE *
- ENDSCAN
- Teaching point
- LOOP / WHILE / UNTIL are general control-flow constructs.

- SCAN is a data-aware loop specialized for table traversal.

Status

- implemented=no; supported=yes

Syntax

- LOOPS

Provenance

- Topic key: ED|LOOPS
- Included HELP rows: 32
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LOWER

LOWER

- Catalog/topic: FOX / LOWER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert <cExpression> to lower-case.

Status

- implemented=yes; supported=yes

Syntax

- LOWER(<cExpression>)

Provenance

- Topic key: FOX|LOWER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

LTRIM

LTRIM

- Catalog/topic: FOX / LTRIM
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Remove leading spaces from <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- LTRIM(<cExpression>)

Provenance

- Topic key: FOX|LTRIM
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

MCC

MCC

- Catalog/topic: DOT / MCC
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Load the MCC v32 demo workspace by running DotScript x32 and WORKSPACE LOAD mcc.dtschemas.

- Load the MCC v32 demo workspace as a one-command starter demo.

Status

- implemented=yes; supported=yes

Syntax

- MCC

Usage

- MCC
- MCC USAGE

Note

- MCC prepares and loads the MCC sample workspace for demonstration.
- MCC runs DotScript x32 to set the v32 DBF and INDEX paths.
- MCC then runs WORKSPACE LOAD mcc.dtschemas.
- Equivalent manual sequence is DOTSCRIPT x32, then WORKSPACE LOAD mcc.dtschemas.
- MCC is a convenience command and does not directly open tables or create relations itself.
- Table/session/relation restoration remains owned by WORKSPACE.
- Environment/path setup remains owned by DotScript.
- DO X32 is a command-surface shortcut for DotScript x32; MCC should be documented as using DotScript.

Provenance

- Topic key: DOT|MCC
- Included HELP rows: 14
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

MODEL

MODEL

- Catalog/topic: ED / MODEL
- Status: **supported**
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary How to think about DotTalk++DotTalk++ is best understood as four layers working together:1. Command Layer The CLI accepts commands such as: USE SELECT LIST SET

- How to think about DotTalk++
- DotTalk++ is best understood as four layers working together:
 - 1. Command Layer
 - The CLI accepts commands such as:
 - USE
 - SELECT
 - LIST
 - SET ORDER
 - SET RELATION
 - REL ENUM
 - 2. Data Layer
 - The engine works with:
 - tables
 - records
 - fields
 - indexes
 - relations
 - 3. Logic Layer
 - DotScript and command execution provide:
 - sequential flow
 - decision flow
 - looping flow
 - expression evaluation
 - 4. View / Projection Layer
 - Results are shown through:
 - DISPLAY
 - TUPLE
 - SMARTLIST
 - browser commands
 - A practical mental model
 - Table = stored rows

- Record = one row
- Field = one column value in a row
- Index = alternate ordered path through rows
- Relation = parent-child path between tables
- Tuple = projected logical row, possibly across multiple tables

Status

- implemented=no; supported=yes

Syntax

- SYSTEM MODEL

Provenance

- Topic key: ED|MODEL
- Included HELP rows: 38
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

NAVIGATION

NAVIGATION

- Catalog/topic: ED / NAVIGATION
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Navigation means moving the current record pointer. Common navigation commands TOP BOTTOM SKIP GOTO SEEK FIND LOCATE Important idea Many commands are pointer-sensitive

- Navigation means moving the current record pointer.
- Common navigation commands
- TOP
- BOTTOM
- SKIP
- GOTO
- SEEK
- FIND
- LOCATE
- Important idea
- Many commands are pointer-sensitive.
- They use “current row” as their starting context.
- Teaching point
- Navigation is to a database session what cursor movement is to an editor.

Status

- implemented=no; supported=yes

Syntax

- NAVIGATION

Provenance

- Topic key: ED|NAVIGATION
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

NORMALIZE

NORMALIZE

- Catalog/topic: DOT / NORMALIZE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Normalize/clean an input expression or text (developer utility).

Status

- implemented=yes; supported=yes

Syntax

- NORMALIZE <expr>

Provenance

- Topic key: DOT|NORMALIZE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ORDER

ORDER

- Catalog/topic: ED / ORDER
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Order means the currently active navigation sequence.Examples physical order LNAME order FNAME
orderCommands SET ORDER TO TAG LNAME SET ORDER TO 0 ASCEND DESCEND TO

- Order means the currently active navigation sequence.
- Examples
- physical order
- LNAME order
- FNAME order
- Commands
- SET ORDER TO TAG LNAME
- SET ORDER TO 0
- ASCEND
- DESCEND
- TOP
- BOTTOM
- SEEK

- FIND
- Educational point
- TOP is not an abstract idea.
- TOP means “first row in the current order.”
- So:
- TOP in physical order may differ from
- TOP in indexed order.
- This is one of the most important database-learning moments in DotTalk++.

Status

- implemented=no; supported=yes

Syntax

- ORDER AND NAVIGATION

Provenance

- Topic key: ED|ORDER
- Included HELP rows: 23
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PACK

PACK

- Catalog/topic: DOT / PACK
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Permanently remove deleted records from the current table.

- Physically remove deleted records by rewriting the current DBF; x64 memo tables rebuild both DBF and DTX sidecar with remapped memo ids.

Status

- implemented=yes; supported=yes

Syntax

- PACK

Usage

- PACK USAGE
- PACK

Example

- PACK

Note

- PACK USAGE prints usage before open-table checks.
- PACK rewrites the current DBF with only non-deleted records and closes the table on success.
- PACK supports x64 M(8) memo tables by rebuilding DBF and DTX together.
- Legacy memo tables are refused.
- Index containers must be rebuilt/rebound after PACK.

Provenance

- Topic key: DOT|PACK
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PADC

PADC

- Catalog/topic: FOX / PADC
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Center-pad to length <nLen>.

Status

- implemented=no; supported=yes

Syntax

- PADC(<cExpression>, <nLen>[, <cFill>])

Provenance

- Topic key: FOX|PADL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PADL

PADL

- Catalog/topic: FOX / PADL
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Left-pad to length <nLen>.

Status

- implemented=no; supported=yes

Syntax

- PADL(<cExpression>, <nLen>[, <cFill>])

Provenance

- Topic key: FOX|PADL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PADR

PADR

- Catalog/topic: FOX / PADR
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Right-pad to length <nLen>.

Status

- implemented=no; supported=yes

Syntax

- PADR(<cExpression>, <nLen>[, <cFill>])

Provenance

- Topic key: FOX|PADR
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PREDHELP

PREDHELP

- Catalog/topic: DOT / PREDHELP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Help for predicates/expressions and filtering.

- Report predicate syntax, operators, functions, and filtering examples.

Status

- implemented=yes; supported=yes

Syntax

- PREDHELP [<topic>]

Usage

- PREDHELP
- PREDICATES

Note

- PREDICATES is the registered descriptive alias of PREDHELP.

Alias

- PREDICATES

Provenance

- Topic key: DOT|PREDHELP
- Included HELP rows: 8
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PREDICATE

PREDICATE

- Catalog/topic: ED / PREDICATE
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary A predicate is a true/false expression.Examples GPA > 3.5 LNAME = “TAYLOR” GPA >= 2 AND GPA <= 4Used by IF LOCATE COUNT WHERE SMARTLIST SET FILTERBoolean logi

- A predicate is a true/false expression.
- Examples
- GPA > 3.5
- LNAME = “TAYLOR”
- GPA >= 2 AND GPA <= 4
- Used by
- IF
- LOCATE
- COUNT
- WHERE
- SMARTLIST
- SET FILTER
- Boolean logic words
- AND
- OR
- NOT
- Teaching point
- Predicates are the language of selection.
- They control decisions, filters, and searches.

Status

- implemented=no; supported=yes

Syntax

- PREDICATES AND BOOLEAN LOGIC

Provenance

- Topic key: ED|PREDICATE
- Included HELP rows: 21
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PREDICATES

PREDICATES

- Catalog/topic: DOT / PREDICATES
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary List supported predicates/operators for filtering.

Status

- implemented=yes; supported=yes

Syntax

- PREDICATES

Provenance

- Topic key: DOT|PREDICATES
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PROJECTION

PROJECTION

- Catalog/topic: ED / PROJECTION
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary Projection means choosing which columns/fields to show. Examples TUPLE LNAME,FNAME LIST FIELDS LNAME,FNAME REL ENUM ... TUPLE STUDENTS.SID,TEACHERS.LNAME Educational point Projection

- Projection means choosing which columns/fields to show.
- Examples
- TUPLE LNAME,FNAME
- LIST FIELDS LNAME,FNAME
- REL ENUM ... TUPLE STUDENTS.SID,TEACHERS.LNAME
- Educational point
- Projection is one of the three great data operations:
- selection = which rows
- ordering = in what sequence
- projection = which columns

Status

- implemented=no; supported=yes

Syntax

- PROJECTION

Provenance

- Topic key: ED|PROJECTION
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PROJECTS

PROJECTS

- Catalog/topic: DOT / PROJECTS
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect or manage DotTalk++ project/workflow entries and project-oriented local state.

- List, create, inspect, tree, or delete project skeleton directories under the configured PROJECTS slot.

Status

- implemented=yes; supported=yes

Syntax

- PROJECTS [USAGE|LIST|OPEN <name>|STATUS]

Usage

- PROJECTS
- PROJECTS USAGE
- PROJECTS LIST
- PROJECTS CREATE <name> [DATA|FEATURE|HYBRID]
- PROJECTS INFO <name>
- PROJECTS TREE <name>
- PROJECTS DELETE <name> [CONFIRM]

Example

- PROJECTS
- PROJECTS CREATE demo DATA
- PROJECTS INFO demo
- PROJECTS TREE demo
- PROJECTS DELETE demo
- PROJECTS DELETE demo CONFIRM

Note

- PROJECTS with no arguments lists known projects.
- CREATE writes project skeleton folders and a manifest.
- DELETE is dry-run unless CONFIRM is supplied.
- PROJECTS USAGE prints usage and does not create/delete files.

Provenance

- Topic key: DOT|PROJECTS
- Included HELP rows: 21
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PROPER

PROPER

- Catalog/topic: FOX / PROPER
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert <cExpression> to Proper Case (title case).

Status

- implemented=no; supported=yes

Syntax

- PROPER(<cExpression>)

Provenance

- Topic key: FOX|PROPER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

PSHELL

PSHELL

- Catalog/topic: DOT / PSHELL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Invoke or document the PowerShell/platform-shell helper surface where enabled by the runtime policy.

- PowerShell reference catalog presentation helper used by the PSHELL command.
- Display the PowerShell/PSHELL helper reference and search the curated PowerShell one-liner catalog.

Status

- implemented=yes; supported=yes

Syntax

- PSHELL [USAGE|<command...>]

Usage

- User-visible PSHELL usage is owned by cmd_pshell_help.cpp.
- This file provides show_pshell_help(...) and catalog formatting support.
- PSHELL
- PSHELL USAGE
- PSHELL LIST-CATEGORIES
- PSHELL <category>
- PSHELL <term>

Example

- PSHELL
- PSHELL PYTHON
- PSHELL PY-VEENV-CREATE
- PSHELL CLEAN

Note

- PSHELL is read-only reference output; it does not execute PowerShell.
- Keep command dispatch/usage gating in cmd_pshell_help.cpp.
- PSHELL with no arguments displays the grouped PSHELL reference.
- PSHELL USAGE prints command usage without searching the catalog.
- PSHELL is read-only and does not execute PowerShell commands.

Provenance

- Topic key: DOT|PSHELL
- Included HELP rows: 21
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REBUILD

REBUILD

- Catalog/topic: DOT / REBUILD
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Rebuild table/index/help-related derived state where supported by the current target context.

- Rebuild a CNX container for the current table, using the active CNX or a supplied CNX name/path and clearing TABLE stale state on success.

Status

- implemented=yes; supported=yes

Syntax

- REBUILD [USAGE|ALL|<target>]

Usage

- REBUILD USAGE
- REBUILD
- REBUILD <name-or-path.cnx>

Note

- REBUILD with no arguments uses the current CNX or defaults to <table>.cnx.
- REBUILD requires an open table except for REBUILD USAGE.
- REBUILD prompts to COMMIT dirty TABLE buffers before rebuilding.
- REBUILD refuses to continue if the table remains dirty after COMMIT.
- REBUILD opens the CNX tag directory once for reporting.
- The CNX backend rebuilds all tags in the container in one rebuild call.
- On success, TABLE STALE is cleared for the current area when table buffering is enabled.

Provenance

- Topic key: DOT|REBUILD
- Included HELP rows: 14
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RECALL

RECALL

- Catalog/topic: DOT / RECALL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Unmark the current deleted record when supported by the current table state.

- Clear deleted flags on the current record or selected deleted records.

Status

- implemented=yes; supported=yes

Syntax

- RECALL

Usage

- RECALL USAGE
- RECALL
- RECALL ALL
- RECALL REST
- RECALL NEXT <n>
- RECALL FOR <expr>
- UNDELETE

Example

- RECALL
- RECALL ALL
- RECALL REST

- RECALL NEXT 10
- RECALL FOR LNAME = “SMITH”

Note

- RECALL USAGE prints usage before open-table checks.
- RECALL with no arguments recalls the current record.
- RECALL target selection is deleted-only.
- RECALL rebuilds index entries for recalled records best-effort.
- UNDELETE is the registered compatibility alias of RECALL.

Alias

- UNDELETE

Provenance

- Topic key: DOT|RECALL
- Included HELP rows: 22
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REINDEX

REINDEX

- Catalog/topic: DOT / REINDEX
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Rebuild index files for the current table (or all open tables).

- Canonical rebuild dispatcher for INX, CNX, CDX/LMDB, and student index families, choosing a default family by table flavor when no family is given.

Status

- implemented=yes; supported=yes

Syntax

- REINDEX [ALL]

Usage

- REINDEX USAGE
- REINDEX
- REINDEX INX
- REINDEX INX <tagfile>
- REINDEX CNX
- REINDEX CNX <name-or-path.cnx>
- REINDEX CDX
- REINDEX CDX YES
- REINDEX CDX AUTO
- REINDEX CDX NOPROMPT
- REINDEX CDX CLEAN
- REINDEX CDX FORCE
- REINDEX CDX QUIET

- REINDEX SIX
- REINDEX SIX <tagfile>
- REINDEX SCX
- REINDEX SCX <tagfile>
- REINDEX ALL
- REINDEX CUSTOM
- REINDEX <tagfile>

Note

- REINDEX with no arguments chooses the default family by open table flavor.
- v64-like tables default to CDX through BUILDLMDB.
- v32-like tables default to INX.
- With no table open, the fallback default is CDX.
- REINDEX INX rebuilds a legacy single-tag INX file.
- REINDEX CNX delegates to REBUILD.
- REINDEX CDX delegates to BUILDLMDB.
- REINDEX ALL excludes SIX and SCX by design.
- REINDEX CUSTOM runs SIX and SCX student families.
- REINDEX <tagfile> is treated as REINDEX INX <tagfile> for compatibility.
- Dirty TABLE buffers may prompt for COMMIT before supported rebuild paths.

Provenance

- Topic key: DOT|REINDEX
- Included HELP rows: 35
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REL

REL

- Catalog/topic: DOT / REL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Relations engine commands (native DotTalk++ relation graph). Subcommands: REL LIST REL LIST ALL REL REFRESH REL ENUM [LIMIT <n>] <child1> [<child2> ...] TUPLE <tuple-expr> REL SAV

- Manage and inspect the DotTalk++ relation graph
- Dispatch relation list, refresh, join, enumeration, persistence, add, and clear operations.
- Relations engine commands (native DotTalk++ relation graph).
- Subcommands:
- REL LIST
- REL LIST ALL
- REL REFRESH
- REL ENUM [LIMIT <n>] <child1> [<child2> ...] TUPLE <tuple-expr>
- REL SAVE [path] | REL SAVE AS <dataset>
- REL LOAD [path] | REL LOAD AS <dataset>
- REL ADD <parent> <child> ON <field>[,<field>...]
- REL CLEAR <parent>|ALL
- Examples:
- REL ADD STUDENTS ENROLL ON SID
- REL ADD ENROLL CLASSES ON CLS_ID
- REL ENUM LIMIT 10 ENROLL CLASSES TASSIGN TEACHERS TUPLE ;

- STUDENTS.SID,STUDENTS.LNAME,ENROLL.CLS_ID,CLASSES.CID,TEACHERS.LNAME
- Conceptual Model:
- REL maintains a directed relation graph between work areas.
- REL ADD defines edges.
- REL REFRESH recalculates relation state.
- REL ENUM traverses the graph and emits tuple rows.
- Traversal Behavior:
- REL ENUM performs depth-first traversal of the relation chain.
- Each successful path produces one tuple row.
- Example chain:
- STUDENTS -> ENROLL -> CLASSES -> TASSIGN -> TEACHERS
- Matching:
- Current implementation uses same-name field equality:
- parent.FIELD == child.FIELD
- Notes:
- REL is the native DotTalk++ relation system.
- FoxPro-style SET RELATION commands map into this backend.
- REL ENUM is the row enumeration engine used by relational
- browsers and tuple views.

Status

- implemented=yes; supported=yes

Syntax

- REL LIST
- REL LIST ALL
- REL REFRESH
- REL ADD <parent> <child> ON <field>
- REL ADD <parent> <child> ON <parentField> TO <childField>
- REL CLEAR <parent>
- REL CLEAR ALL
- REL ENUM [LIMIT <n>] <path...> TUPLE <projection>
- REL SAVE [path]
- REL LOAD [path]
- REL <subcommand> ...

Usage

- REL
- REL USAGE
- REL LIST [ALL]
- REL REFRESH
- REL JOIN [LIMIT <n>] [<child1> <child2> ...] TUPLE <expr>
- REL ENUM [LIMIT <n>] [<child1> <child2> ...] TUPLE <expr>
- REL SAVE [path] | REL SAVE AS <dataset>
- REL LOAD [path] | REL LOAD AS <dataset>
- REL ADD <parent> <child> ON <field>[,<field>...]
- REL ADD <parent> <child> ON <parent_field> TO <child_field>
- REL CLEAR <parent>|ALL

Example

- REL LIST
- REL ADD STUDENTS ENROLL ON SID
- REL ADD SYS_CMD SYS_SUBCMD ON CAN_NAME TO PARENT
- REL REFRESH

- REL ENUM LIMIT 10 ENROLL CLASSES TUPLE STUDENTS.SID,CLASSES.CID

Note

- REL is the native relation backend
- FoxPro-style SET RELATION syntax routes into this model where implemented
- REL ENUM traverses relation paths and emits tuple projections
- REL forwards each subcommand to the owning relation handler.
- REL ADD and REL CLEAR mutate relation definitions; REL REFRESH refreshes relation state.

Provenance

- Topic key: DOT|REL
- Included HELP rows: 68
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REL_ENUM

REL ENUM

- Catalog/topic: DOT / REL ENUM
- Status: **supported**
- Implemented/supported: F / T
- Primary/confidence: DOTREF / CATALOG

Summary Traverse a relation chain and emit tuple rows.Syntax: REL ENUM [LIMIT <n>] <child1> [<child2> ...] TUPLE <tuple-expr>Example: REL ENUM LIMIT 5 ENROLL CLASSES TASSIGN TEACHERS TUPLE

- Traverse a relation chain and emit tuple rows.
- Syntax:
- REL ENUM [LIMIT <n>] <child1> [<child2> ...] TUPLE <tuple-expr>
- Example:
- REL ENUM LIMIT 5 ENROLL CLASSES TASSIGN TEACHERS
- TUPLE STUDENTS.SID,STUDENTS.LNAME,TEACHERS.LNAME
- Execution Model:
- 1. Current selected area is the root.
- 2. Each child token represents the next relation edge.
- 3. Traversal expands matches depth-first.
- Example traversal:
- STUDENTS
- -> ENROLL
- -> CLASSES
- -> TASSIGN
- -> TEACHERS
- Each complete path emits one tuple row.
- LIMIT:
- LIMIT applies to emitted rows, not intermediate matches.

- Projection:
- The TUPLE clause defines which fields are emitted.
- Used By:
- SIMPLEBROWSER
- SMARTBROWSER
- relational tuple views.

Status

- implemented=no; supported=yes

Syntax

- REL ENUM [LIMIT <n>] <path...> TUPLE <tuple-expr>

Provenance

- Topic key: DOT|REL ENUM
- Included HELP rows: 27
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REL_REFRESH

REL_REFRESH

- Catalog/topic: DOT / REL_REFRESH
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Refresh relation state for the current workspace through the native REL backend.

- Refresh active relations for the current parent area.

Status

- implemented=yes; supported=yes

Syntax

- REL_REFRESH

Usage

- REL_REFRESH

Note

- This command currently has no usage-only branch; invoking it performs the refresh.
- Use REL USAGE or RELATIONS USAGE for non-mutating documentation access.

Provenance

- Topic key: DOT|REL_REFRESH
- Included HELP rows: 7
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RELATION

RELATION

- Catalog/topic: ED / RELATION
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary A relation connects a parent table to a child table. Example STUDENTS -> ENROLL ON SID ENROLL -> CLASSES ON CLS_ID Meaning If the current STUDENTS row has SID = 50000020, the child

- Manage and inspect the DotTalk++ relation graph
- A relation connects a parent table to a child table.
- Example
- STUDENTS -> ENROLL ON SID
- ENROLL -> CLASSES ON CLS_ID
- Meaning
- If the current STUDENTS row has SID = 50000020,
- the child ENROLL rows with that SID are related rows.
- Public surfaces
- SET RELATION TO SID INTO ENROLL
- REL ADD STUDENTS ENROLL ON SID
- REL LIST
- REL LIST ALL
- REL REFRESH
- Teaching point
- Relations are the backbone of multi-table thinking.
- They let a current parent row lead you into matching child rows.

Status

- implemented=no; supported=yes

Syntax

- REL LIST
- REL LIST ALL
- REL REFRESH
- REL ADD <parent> <child> ON <field>
- REL ADD <parent> <child> ON <parentField> TO <childField>
- REL CLEAR <parent>
- REL CLEAR ALL
- REL ENUM [LIMIT <n>] <path...> TUPLE <projection>
- REL SAVE [path]
- REL LOAD [path]
- RELATIONS

Example

- REL LIST

- REL ADD STUDENTS ENROLL ON SID
- REL ADD SYS_CMD SYS_SUBCMD ON CAN_NAME TO PARENT
- REL REFRESH
- REL ENUM LIMIT 10 ENROLL CLASSES TUPLE STUDENTS.SID,CLASSES.CID

Note

- REL is the native relation backend
- FoxPro-style SET RELATION syntax routes into this model where implemented
- REL ENUM traverses relation paths and emits tuple projections

Provenance

- Topic key: ED|RELATION
- Included HELP rows: 37
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RELATIONS

RELATIONS

- Catalog/topic: DOT / RELATIONS
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Compatibility-facing relation listing surface backed by the native REL engine.Examples: RELATIONS
RELATIONS ALLNotes: RELATIONS and REL LIST point at the same relation-state reporting

- Manage and inspect the DotTalk++ relation graph
- Inspect and manage active relation definitions, relation files, and relation enumeration helpers.
- Compatibility-facing relation listing surface backed by the native REL engine.
- Examples:
- RELATIONS
- RELATIONS ALL
- Notes:
- RELATIONS and REL LIST point at the same relation-state reporting lane.
- Prefer REL for the canonical DotTalk++ relation command family.

Status

- implemented=yes; supported=yes

Syntax

- REL LIST
- REL LIST ALL
- REL REFRESH
- REL ADD <parent> <child> ON <field>
- REL ADD <parent> <child> ON <parentField> TO <childField>
- REL CLEAR <parent>
- REL CLEAR ALL
- REL ENUM [LIMIT <n>] <path...> TUPLE <projection>
- REL SAVE [path]
- REL LOAD [path]
- RELATIONS [USAGE|ALL]

Usage

- RELATIONS
- RELATIONS USAGE
- RELATIONS ALL
- SET RELATIONS
- SET RELATIONS USAGE
- SET RELATIONS ADD <parent> <child> ON f1[,f2...] [TO child_f1[,child_f2...]]
- SET RELATIONS CLEAR <parent|ALL>

Example

- REL LIST
- REL ADD STUDENTS ENROLL ON SID
- REL ADD SYS_CMD SYS_SUBCMD ON CAN_NAME TO PARENT
- REL REFRESH
- REL ENUM LIMIT 10 ENROLL CLASSES TUPLE STUDENTS.SID,CLASSES.CID
- RELATIONS
- RELATIONS ALL
- SET RELATIONS ADD STUDENTS ENROLL ON SID
- SET RELATIONS CLEAR ALL

Note

- REL is the native relation backend
- FoxPro-style SET RELATION syntax routes into this model where implemented
- REL ENUM traverses relation paths and emits tuple projections
- RELATIONS USAGE prints usage and does not inspect or mutate relation state.
- SET RELATIONS USAGE prints usage and does not mutate the relation graph.
- SET RELATIONS ADD/CLEAR mutate relation definitions.
- RELATIONS ALL reports a recursive tree rooted at the current parent.

Alias

- REL_LIST

Provenance

- Topic key: DOT|RELATIONS
- Included HELP rows: 45
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REPLACE

REPLACE

- Catalog/topic: DOT / REPLACE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Replace one field in the current record using table-buffer and memo-aware semantics.

- Replace one field in the current record by field name or field index, preserving RHS expression evaluation, type validation, memo conversion, and table-buffer semantics.

Status

- implemented=yes; supported=yes

Syntax

- REPLACE <field> WITH <value>

Usage

- REPLACE USAGE
- REPLACE <field_index> WITH <value>
- REPLACE <field_name> WITH <value>

Example

- REPLACE LNAME WITH “Smith”
- REPLACE 3 WITH TODAY
- REPLACE NOTES WITH “updated memo text”

Note

- REPLACE requires an open table and a current record.
- REPLACE resolves fields by standard field index/name rules.
- RHS values pass through the expression/RHS evaluator and legacy string/date function handling.
- X64 memo text is converted into stored object-id text before DBF storage.
- Field values are validated and normalized before storage.
- When TABLE buffering is ON, REPLACE records a buffered field change and marks the field stale/dirty.
- When TABLE buffering is OFF, REPLACE writes immediately through DbArea storage.
- COMMIT owns durable application of buffered table changes.
- REPLACE is a table-data mutation command; do not classify it as read-only.

Provenance

- Topic key: DOT|REPLACE
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

REPLICATE

REPLICATE

- Catalog/topic: FOX / REPLICATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Repeat <cExpression> <nTimes> times.

Status

- implemented=yes; supported=yes

Syntax

- REPLICATE(<cExpression>, <nTimes>)

Provenance

- Topic key: FOX|REPLICATE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RETRO

RETRO

- Catalog/topic: DOT / RETRO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display ASCII-safe retro computer and system splash screens.

- Display retro computer/system splash screens with system-specific terminal profiles.

Status

- implemented=yes; supported=yes

Syntax

- RETRO [USAGE|LIST|SHOW <system>|<system>|HELP]

Usage

- RETRO USAGE
- RETRO LIST
- RETRO LIST LONG
- RETRO STYLES
- RETRO MODES
- RETRO SHOW <system>
- RETRO SHOW <system> NATIVE
- RETRO SHOW <system> ASCII
- RETRO SHOW <system> LEGACY
- RETRO SHOW <system> STYLE <style>
- RETRO SHOW <system> NOCLEAR
- RETRO SHOW <system> NOCAPTION
- RETRO <system>
- RETRO <system> INFO
- RETRO HELP

Example

- RETRO C64
- RETRO C64 ASCII NOCLEAR
- RETRO C64 STYLE GREEN
- RETRO IBMPC NATIVE
- RETRO IBMPC STYLE MDA
- RETRO VT100
- RETRO AMIGA NATIVE
- RETRO GBC NATIVE
- RETRO PS2 NATIVE
- RETRO XBOX NATIVE
- RETRO C64 INFO

Note

- NATIVE uses a system-specific profile and default ANSI style.
- ASCII uses the system-specific plate without ANSI color.
- LEGACY uses the older framed catalog plate where available.
- STYLE overrides the NATIVE default color treatment.
- NOCLEAR is useful for DotScript logs, tests, and transcript capture.
- RETRO writes console output only and does not mutate table data.

Provenance

- Topic key: DOT|RETRO
- Included HELP rows: 36
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RIGHT

RIGHT

- Catalog/topic: FOX / RIGHT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return the right-most <nChars> characters of <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- RIGHT(<cExpression>, <nChars>)

Provenance

- Topic key: FOX|RIGHT
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ROLLBACK

ROLLBACK

- Catalog/topic: DOT / ROLLBACK
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Discard staged TABLE/buffered changes without committing them; operational behavior depends on the current buffering state.

- Discard buffered/uncommitted table changes for the current area or all areas.

Status

- implemented=yes; supported=yes

Syntax

- ROLLBACK [USAGE|HELP|?]

Usage

- ROLLBACK USAGE
- ROLLBACK
- ROLLBACK ALL

Example

- ROLLBACK
- ROLLBACK ALL

Note

- ROLLBACK USAGE returns before modifying buffer state.
- ROLLBACK without arguments clears buffered state for the current area.
- ROLLBACK ALL clears buffered state across all areas.

Provenance

- Topic key: DOT|ROLLBACK
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RTRIM

RTRIM

- Catalog/topic: FOX / RTRIM
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Remove trailing spaces from <cExpression>.

Status

- implemented=yes; supported=yes

Syntax

- RTRIM(<cExpression>)

Provenance

- Topic key: FOX|RTRIM
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RULE

RULE

- Catalog/topic: DOT / RULE
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect or manage validation-rule metadata and rule-oriented helper workflows.

- Inspect rule catalog paths, bindings, and field constraints for the current work area.

Status

- implemented=yes; supported=yes

Syntax

- RULE [USAGE|<args...>]

Usage

- RULE
- RULE USAGE
- RULE STATUS
- RULE SHOW <field|ALL>
- RULE LIST
- RULE PATHS

Example

- RULE
- RULE STATUS
- RULE SHOW GPA
- RULE SHOW ALL
- RULE LIST
- RULE PATHS

Note

- RULE with no arguments reports rule status.
- RULE USAGE prints usage and does not require an open table.
- RULE is diagnostic/read-only; it does not create, edit, or bind rules.

Provenance

- Topic key: DOT|RULE
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

RUN

RUN

Status notice: **PARTIAL**; supported=F. Treat this page as limited or review-required evidence.

- Catalog/topic: FOX / RUN
- Status: **partial**

- Implemented/supported: F / F
- Primary/confidence: FOXREF / CATALOG

Summary Run an OS command (FoxPro).

Status

- implemented=no; supported=no

Syntax

- RUN /N <command> | RUN <file>

Provenance

- Topic key: FOX|RUN
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SCAN

SCAN

- Catalog/topic: DOT / SCAN
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Iterate records using a SCAN block (record loop).

- Buffer and execute a SCAN...ENDSCAN record loop over the current logical rowset.

Status

- implemented=yes; supported=yes

Syntax

- SCAN [ALL|DELETED] [FOR <pred>] [WHILE <pred>] [NEXT <n>|REST]

Usage

- SCAN
- SCAN USAGE
- SCAN FOR <expr>
- ENDSCAN
- ENDSCAN USAGE

Note

- SCAN with no arguments starts buffering a scan block on the current area.
- SCAN FOR <expr> adds a predicate to the scan loop.
- ENDSCAN executes buffered body lines through the canonical command executor.
- Deleted records and active SET FILTER visibility are honored by the scan gate.
- Active order traversal uses shared order_iterator when available, with physical fallback.
- ENDSCAN restores the user's cursor best-effort after execution.

Provenance

- Topic key: DOT|SCAN
- Included HELP rows: 15
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SCAN_BUFFER

SCAN_BUFFER

- Catalog/topic: DOT / SCAN_BUFFER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic helper surface for inspecting or testing buffered SCAN/table-iteration behavior.

Status

- implemented=yes; supported=yes

Syntax

- SCAN_BUFFER [USAGE|<args...>]

Provenance

- Topic key: DOT|SCAN_BUFFER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SCHEMAS

SCHEMAS

- Catalog/topic: DOT / SCHEMAS
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Deprecated compatibility shim for WORKSPACE. Current mapping: SCHEMAS -> WORKSPACE
SCHEMAS OPEN <arg> -> WORKSPACE OPEN <arg> SCHEMAS CLOSE

- Manage live work-area/session state
- Deprecated compatibility shim that routes old SCHEMAS commands to WORKSPACE.
- Deprecated compatibility shim for WORKSPACE.
- Current mapping:
- SCHEMAS -> WORKSPACE
- SCHEMAS OPEN <arg> -> WORKSPACE OPEN <arg>
- SCHEMAS CLOSE -> WORKSPACE CLOSE
- Notes:
- Use WORKSPACE for live work-area/session operations.
- Use DDL for schema/definition operations.
- SCHEMAS remains only so older DotScript files continue to run.

Status

- implemented=yes; supported=yes

Syntax

- WORKSPACE
- WORKSPACE OPEN DBF
- WORKSPACE OPEN <dir>
- WORKSPACE ADD <file.dbf>
- WORKSPACE CLOSE
- WORKSPACE SAVE <name>
- WORKSPACE LOAD <name>
- SCHEMAS [OPEN <DBF|dir>|CLOSE]

Usage

- SCHEMAS
- SCHEMAS USAGE
- SCHEMAS OPEN <arg>
- SCHEMAS CLOSE

Example

- WORKSPACE
- WORKSPACE CLOSE
- WORKSPACE OPEN DBF
- WORKSPACE ADD students
- WORKSPACE SAVE mcc
- WORKSPACE LOAD mcc.dtschemas

Note

- WORKSPACE owns live areas, aliases, orders, and relation/session layout
- WORKSPACE OPEN replaces area membership; WORKSPACE ADD preserves existing areas
- DDL owns schema/definition work
- SCHEMAS remains a compatibility shim for older scripts
- SCHEMAS with no arguments still routes to WORKSPACE list behavior.
- SCHEMAS OPEN <arg> routes to WORKSPACE OPEN <arg>.
- SCHEMAS CLOSE routes to WORKSPACE CLOSE.
- SCHEMAS USAGE prints compatibility guidance and does not route to WORKSPACE.
- WORKSPACE owns live area/session behavior; DDL owns schema/definition work.

Provenance

- Topic key: DOT|SCHEMAS
- Included HELP rows: 39
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SCRIPT

SCRIPT

- Catalog/topic: ED / SCRIPT
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary DotScript is the scripting surface of DotTalk++.Purpose - automate repetitive commands - build regression tests - teach execution flow - preserve reproducible scenariosWhy it matte

- DotScript is the scripting surface of DotTalk++.
- Purpose
 - automate repetitive commands
 - build regression tests
 - teach execution flow
 - preserve reproducible scenarios
- Why it matters educationally
- DotScript turns interactive use into programming.
- DotScript demonstrates
 - sequential execution
 - decision blocks
 - loops
 - command composition
 - testing discipline

Status

- implemented=no; supported=yes

Syntax

- DOTSCRIPT

Provenance

- Topic key: ED|SCRIPT
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SCX

SCX

- Catalog/topic: DOT / SCX
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Student/local SCX index-file lab command for creating, tagging, building, listing, and inspecting SCX index files.

Status

- implemented=yes; supported=yes

Syntax

- SCX [USAGE|CREATE <file>|ADDTAG <file> <name> FIELD <n> [DESC]|BUILD <file>|TAGS <file>|INFO <file>]

Usage

- SCX USAGE
- SCX CREATE <file>
- SCX ADDTAG <file> <name> FIELD <n>
- SCX ADDTAG <file> <name> FIELD <n> DESC
- SCX BUILD <file>
- SCX TAGS <file>
- SCX INFO <file>

Note

- SCX with no arguments prints usage.
- CREATE writes a new SCX container/file.
- ADDTAG mutates SCX tag metadata.
- BUILD builds SCX contents from the current area.
- TAGS and INFO inspect SCX metadata.
- SCX is separate from the ordinary command-surface CNX/CDX/LMDB abstractions.

Provenance

- Topic key: DOT|SCX
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SECHO

SECHO

- Catalog/topic: DOT / SECHO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Scripted/student echo helper for emitting text from teaching, demo, or test scripts.

Status

- implemented=yes; supported=yes

Syntax

- SECHO [USAGE|<text...>]

Provenance

- Topic key: DOT|SECHO
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SECURITY

SECURITY

- Catalog/topic: DOT / SECURITY
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Inspect DotTalk++ security policy/runtime rules and manage the current shell-session role identity and assignment view.

- Display x64Base security policy/runtime diagnostics or run built-in security self-tests.

Status

- implemented=yes; supported=yes

Syntax

- SECURITY [USAGE|SHOW|SELFTEST|RUNTIME|LOGIN <role> [AS <worker>]]|WHOAMI|ASSIGNMENTS|LOGOUT

Usage

- SECURITY USAGE
- SECURITY SHOW
- SECURITY SELFTEST
- SECURITY RUNTIME
- SECURITY LOGIN <DEVELOPER|TEACHER|STUDENT> [AS <worker>]
- SECURITY WHOAMI
- SECURITY ASSIGNMENTS
- SECURITY LOGOUT

Note

- SECURITY with no arguments prints usage.
- SHOW displays the active policy and profile roots.
- SELFTEST runs built-in security tests.
- RUNTIME describes runtime enforcement rules.
- LOGIN establishes a role/session identity for the current shell session.
- WHOAMI reports the active shell-session role identity.
- ASSIGNMENTS reports the assignment lane bound to the active role.
- LOGOUT clears the active shell-session role identity.
- SECURITY does not mutate table data.

Provenance

- Topic key: DOT|SECURITY
- Included HELP rows: 21
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SEEK

SEEK

- Catalog/topic: DOT / SEEK
- Status: **supported**
- Implemented/supported: T / T

- Primary/confidence: DOTREF / CATALOG

Summary Seek through the active order or by scanning a requested field.

- Seek an exact value using the active index/order
- Seek a value through the active order/tag or by scanning a specified field.

Status

- implemented=yes; supported=yes

Syntax

- SEEK <expr>
- SEEK <expr> IN <alias>
- SEEK <value> [IN <field>] | SEEK <field> = <value>

Usage

- SEEK USAGE
- SEEK <value> IN <field> [TRACE ON|OFF]
- SEEK <field> = <value> [TRACE ON|OFF]
- SEEK <field> <value> [TRACE ON|OFF]
- SEEK <value>
- SEEK TRACE ON
- SEEK TRACE OFF

Example

- SEEK "SMITH"
- SEEK 1001
- SEEK DTOS(DATE())

Note

- Requires an active table and normally an active order/index
- Successful SEEK intentionally moves the current record pointer
- Command wording should stay at the CDX/tag/order level; LMDB is a backend detail
- SEEK USAGE works without an open table.
- Bare SEEK with no open table preserves existing behavior and prints (empty).
- SEEK <value> uses the active order/tag when one is set.
- SET NEAR affects near-match reporting policy.
- SEEK may temporarily move the cursor while searching and leaves the cursor on a found/near match.

Warning

- Do not document direct LMDB path parsing as public SEEK behavior

Provenance

- Topic key: DOT|SEEK
- Included HELP rows: 26
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SELECT

SELECT

- Catalog/topic: DOT / SELECT
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Select the active DotTalk++ work area by number or logical name.

- Select the current work area by numeric slot or by work-area/table name.

Status

- implemented=yes; supported=yes

Syntax

- SELECT <area-or-alias>

Usage

- SELECT USAGE
- SELECT <n>
- SELECT <name>
- SELECT <table.dbf>

Note

- SELECT with no arguments prints usage with the current valid slot range.
- SELECT USAGE prints usage and does not change the current area.
- Numeric selection uses the current workarea slot count.
- Name selection matches workarea labels and open DBF base names case-insensitively.
- SELECT mutates current-area/session state but does not mutate table data.

Provenance

- Topic key: DOT|SELECT
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SFTP

SFTP

- Catalog/topic: DOT / SFTP
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary File-transfer helper surface for SFTP-oriented workflows where enabled by local policy.

- Wrap the system OpenSSH sftp client for LS, GET, and PUT file transfer.

Status

- implemented=yes; supported=yes

Syntax

- SFTP [USAGE|<args...>]

Usage

- SFTP USAGE

Note

- SFTP USAGE prints usage and does not start the sftp client.
- This command stages a temporary sftp batch file and invokes the system sftp client.
- Password embedding in URLs is deliberately not supported.
- Set DOTTALK_ALLOW_HOST_COMMANDS=1 and DOTTALK_ALLOW_NETWORK=1 to enable transfer.

Provenance

- Topic key: DOT|SFTP
- Included HELP rows: 9
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SHELLO

SHELLO

- Catalog/topic: DOT / SHELLO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Small shell/student hello demonstration command for teaching command wiring and output.

Status

- implemented=yes; supported=yes

Syntax

- SHELLO [USAGE|<text...>]

Provenance

- Topic key: DOT|SHELLO
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SHOWINI

SHOWINI

- Catalog/topic: DOT / SHOWINI
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display DotTalk++ initialization/configuration files and resolved startup settings.

- Display a table-specific .ini file, either derived from the current table or from an explicit file/path.

Status

- implemented=yes; supported=yes

Syntax

- SHOWINI [USAGE|SYSTEM|USER|ALL]

Usage

- SHOWINI
- SHOWINI USAGE
- SHOWINI <table-or-ini>
- SHOWINI PATH <ini-file>

Example

- SHOWINI
- SHOWINI students
- SHOWINI students.ini

Note

- SHOWINI with no arguments derives the .ini path from the current table.
- SHOWINI USAGE prints usage before open-table checks or file reads.
- SHOWINI reads .ini files and prints parsed sections/keys; it does not write files.

Provenance

- Topic key: DOT|SHOWINI
- Included HELP rows: 14
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SHUTDOWN

SHUTDOWN

- Catalog/topic: DOT / SHUTDOWN
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Run DotTalk++ shutdown processing, including shutdown.ini when present.

- Run the optional shutdown.ini script from the executable directory.

Status

- implemented=yes; supported=yes

Syntax

- SHUTDOWN

Usage

- SHUTDOWN
- SHUTDOWN USAGE

Note

- SHUTDOWN with no arguments looks for shutdown.ini beside the executable and executes it when present.
- SHUTDOWN USAGE prints usage and does not execute shutdown.ini.
- Each non-empty shutdown.ini line is executed through the shell command executor.
- UTF-8 BOM and trailing carriage returns are handled.
- SHUTDOWN may indirectly mutate data, session state, or files depending on script contents.

Provenance

- Topic key: DOT|SHUTDOWN
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SIX

SIX

- Catalog/topic: DOT / SIX
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Experimental or compatibility index-related surface for SIX-style indexing concepts and diagnostics.

Status

- implemented=yes; supported=yes

Syntax

- SIX [USAGE|<args...>]

Provenance

- Topic key: DOT|SIX
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SKIP

SKIP

- Catalog/topic: DOT / SKIP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Move relative to the current record in the active work area.

- Move the current work-area cursor forward or backward
- Move the current work-area cursor forward or backward using filter-aware navigation selection.

Status

- implemented=yes; supported=yes

Syntax

- SKIP
- SKIP <n>
- SKIP +<n>
- SKIP -<n>
- SKIP [<n>]

Usage

- SKIP
- SKIP USAGE
- SKIP <n>

Example

- SKIP
- SKIP 5
- SKIP -1

Note

- Default movement is one record forward
- Uses active logical order when one is active
- SKIP with no arguments moves forward one logical record.
- SKIP <n> moves forward when n is positive and backward when n is negative.
- SKIP 0 rereads the current record.
- SKIP requires an open table except for SKIP USAGE.
- Navigation uses the shared filter-aware selector.
- SKIP mutates cursor position but does not mutate table data.

Provenance

- Topic key: DOT|SKIP
- Included HELP rows: 23
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SMARTLIST

SMARTLIST

- Catalog/topic: DOT / SMARTLIST
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary LIST-style output with SQL-grade filtering (order-aware). Examples: SMARTLIST 20 FOR gpa >= 3.5 SMARTLIST ALL FOR lname LIKE "SMI%" Notes: Evaluat

- Order-aware list output with predicate support
- Filter-aware, order-aware table listing with optional projections, tuple output, debug tracing, deleted-record modes, and predicate filtering.
- LIST-style output with SQL-grade filtering (order-aware).
- Examples:

- SMARTLIST 20 FOR gpa >= 3.5
- SMARTLIST ALL FOR lname LIKE “SMI%”
- Notes:
- Evaluates filters using the expression/predicate pipeline.

Status

- implemented=yes; supported=yes

Syntax

- SMARTLIST
- SMARTLIST <n>
- SMARTLIST ALL
- SMARTLIST DELETED
- SMARTLIST FOR <expr>
- SMARTLIST DEBUG
- SMARTLIST [ALL|DELETED|<n>] [DEBUG] [FOR <expr>]

Usage

- SMARTLIST
- SMARTLIST USAGE
- SMARTLIST <fields>
- SMARTLIST ALL
- SMARTLIST <limit>
- SMARTLIST NEXT <n>
- SMARTLIST FIRST <n>
- SMARTLIST DELETED
- SMARTLIST DEBUG
- SMARTLIST TUPLES
- SMARTLIST FOR <pred>

Example

- SMARTLIST
- SMARTLIST 20
- SMARTLIST ALL FOR GPA >= 3.5
- SMARTLIST FOR LNAME LIKE “SMI%”

Note

- Preferred listing command for user-facing ordered output
- Respects active order and active filter where available
- Uses expression/predicate services rather than ad-hoc string matching
- SMARTLIST requires an open table except for SMARTLIST USAGE.
- SMARTLIST with no arguments preserves existing behavior and prints usage before continuing with default listing.
- Field projections are comma-separated.
- ALL removes the output limit.
- NEXT and FIRST limit scan scope.
- DELETED selects deleted records.
- TUPLES emits tuple bridge output.
- DEBUG emits order/filter diagnostics.
- FOR applies predicate filtering.
- SMARTLIST restores the original cursor best-effort after listing.
- SMARTLIST is read-only for table data.

Provenance

- Topic key: DOT|SMARTLIST
- Included HELP rows: 45
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SPACE

SPACE

- Catalog/topic: FOX / SPACE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return a character string containing <nSpaces> spaces.

Status

- implemented=yes; supported=yes

Syntax

- SPACE(<nSpaces>)

Provenance

- Topic key: FOX|SPACE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SQL

SQL

- Catalog/topic: DOT / SQL
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Execute an SQL statement using the configured SQL engine.

- Evaluate SQL-like COUNT/FOR predicates over the current DBF work area.

Status

- implemented=yes; supported=yes

Syntax

- SQL <statement>

Usage

- SQL USAGE
- SQL [COUNT] [ALL|DELETED] [FOR <expr> | <expr>] [VERBOSE]

Example

- SQL COUNT
- SQL COUNT ALL
- SQL COUNT DELETED
- SQL COUNT FOR GPA >= 3.0
- SQL LNAME = "SMITH"
- SQL VERBOSE COUNT FOR GPA >= 3.0

Note

- SQL USAGE prints usage before open-table checks.
- SQL reads records and may temporarily move the cursor.
- SQL does not mutate table data.

Provenance

- Topic key: DOT|SQL
- Included HELP rows: 15
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SQLERASE

SQLERASE

- Catalog/topic: DOT / SQLERASE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary SQL helper surface for erase/drop/delete-style SQL maintenance workflows where enabled.

- Mark records deleted using SQL-like ERASE FROM <table> WHERE <expr> syntax.

Status

- implemented=yes; supported=yes

Syntax

- SQLERASE [USAGE|<args...>]

Usage

- SQLERASE USAGE
- SQLERASE FROM <table> WHERE <expr>

Example

- SQLERASE FROM STUDENTS WHERE SID = 1001
- SQLERASE FROM STUDENTS WHERE GPA < 1.0

Note

- SQLERASE USAGE prints usage before open-table checks.
- WHERE is required to reduce accidental destructive operations.
- SQLERASE mutates table data by marking matching records deleted.

Provenance

- Topic key: DOT|SQLERASE
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SQLHELP

SQLHELP

- Catalog/topic: DOT / SQLHELP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Show SQL-oriented help and guidance for DotTalk++ SQL bridge workflows.

- Display or search the SQL helper/reference catalog.

Status

- implemented=yes; supported=yes

Syntax

- SQLHELP [USAGE|<topic>]

Usage

- SQLHELP
- SQLHELP USAGE
- SQLHELP LIST-CATEGORIES
- SQLHELP <category>
- SQLHELP <term>

Example

- SQLHELP
- SQLHELP INDEXING
- SQLHELP CREATE-INDEX
- SQLHELP LIST-CATEGORIES

Note

- SQLHELP with no arguments displays the grouped SQL reference.
- SQLHELP USAGE prints command usage without searching the catalog.
- SQLHELP is read-only and does not execute SQL.

Provenance

- Topic key: DOT|SQLHELP
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SQLITE

SQLITE

- Catalog/topic: DOT / SQLITE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary SQLite integration. Examples: `SQLITE DB datasql_regression.sqlite` `SQLITE EXEC CREATE TABLE t(x INT)` `SQLITE EXEC INSERT INTO t(x) VALUES (1)` `SQ`

- Thin SQLite command wrapper for status, connection management, Bible seed helpers, metadata inspection, SELECT queries, and EXEC statements.
- SQLite integration.
- Examples:
 - `SQLITE DB datasql_regression.sqlite`
 - `SQLITE EXEC CREATE TABLE t(x INT)`
 - `SQLITE EXEC INSERT INTO t(x) VALUES (1)`
 - `SQLITE SELECT * FROM t`
- Notes:
 - Used for regression testing and DBF SQL bridging experiments.

Status

- implemented=yes; supported=yes

Syntax

- `SQLITE <subcommand> ...`

Usage

- `SQLITE`
- `SQLITE USAGE`
- `SQLITE STATUS`
- `SQLITE CWD`
- `SQLITE PWD`
- `SQLITE VERSION`
- `SQLITE OPEN <file>`

Note

- `SQLITE` with no arguments reports connection status and brief usage.
- `SQLITE USAGE`, `HELP`, and question mark print detailed usage.
- `OPEN` and `DB` connect to a SQLite database and create it if needed.
- `BIBLE` and `BIBLECHECK` open/check the canonical Bible seed database when found.
- `EXEC` runs non-SELECT SQL and may mutate the external SQLite database.
- `SELECT` prints query rows and caps output for CLI responsiveness.
- `SQLITE` is independent of DBF open/order state.

Provenance

- Topic key: DOT|SQLITE
- Included `HELP` rows: 25
- `HELP` reference run: `MANRUN-20260717T222026Z-28C704E0`
- Disposition run: `MANRUN-20260717T230554Z-DB3F2DC8`
- Authority: candidate_only; publication_authority_claimed=0

SQLSEL

SQLSEL

- Catalog/topic: DOT / SQLSEL
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Execute an SQL SELECT and display results.

- Evaluate SQL-like selection predicates over the current DBF work area.

Status

- implemented=yes; supported=yes

Syntax

- SQLSEL <expr>

Usage

- SQLSEL USAGE
- SQLSEL [COUNT] [ALL|DELETED] [FOR <expr> | <expr>]

Example

- SQLSEL COUNT
- SQLSEL COUNT ALL
- SQLSEL COUNT FOR GPA >= 3.0
- SQLSEL LNAME = "SMITH"

Note

- SQLSEL USAGE prints usage before open-table checks.
- SQLSEL reads records and may temporarily move the cursor.
- SQLSEL does not mutate table data.

Provenance

- Topic key: DOT|SQLSEL
- Included HELP rows: 13
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SQLVER

SQLVER

- Catalog/topic: DOT / SQLVER
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Report SQLite availability and version.

- Report whether SQLite support is available and print the linked SQLite library version when available.

Status

- implemented=yes; supported=yes

Syntax

- SQLVER

Usage

- SQLVER
- SQLVER USAGE

Note

- SQLVER with no arguments reports SQLite availability/version.
- SQLVER is read-only and does not open SQLite databases.

Provenance

- Topic key: DOT|SQLVER
- Included HELP rows: 8
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STATE

STATE

- Catalog/topic: ED / STATE
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: EDREF / CATALOG

Summary State is one of the most important DotTalk++ concepts. Definition State means the current internal condition of the session. Examples of session state - current work area - current table

- State is one of the most important DotTalk++ concepts.
- Definition
- State means the current internal condition of the session.
- Examples of session state
 - – current work area
 - – current table
 - – current record pointer
 - – active order/tag
 - – active filter
 - – relation graph
 - – buffering on/off
 - – path settings
- Why it matters
- Commands do not operate in a vacuum.
- They operate against the current state.

- Example
- SELECT STUDENTS
- SET ORDER TO TAG LNAME
- SEEK TAYLOR
- Here SEEK depends on:
 - – STUDENTS being current
 - – an order being active
 - – the key format of that order
- Teaching point
- Learning DotTalk++ means learning stateful programming.

Status

- implemented=no; supported=yes

Syntax

- ENGINE STATE

Provenance

- Topic key: ED|STATE
- Included HELP rows: 27
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STR

STR

- Catalog/topic: FOX / STR
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert numeric <nExpr> to a character string (optionally controlling width/decimals).

Status

- implemented=yes; supported=yes

Syntax

- STR(<nExpr>[, <nLen>[, <nDec>]])

Provenance

- Topic key: FOX|STR
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STRUCT

STRUCT

- Catalog/topic: DOT / STRUCT
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Display table structure.

- Display the structure of the current table
- Report DBF field structure and index/container information for the current area or all open areas.
- Minimal translation unit reserved for future STRUCT helper code.

Status

- implemented=yes; supported=yes

Syntax

- STRUCT

Usage

- STRUCT
- STRUCT USAGE
- STRUCT INDEX
- STRUCT FIELDS
- STRUCT ALL
- STRUCT ALL INDEX
- STRUCT ALL VERBOSE
- This file intentionally does not export cmd_STRUCT().
- STRUCT command behavior and usage are owned by the actual STRUCT command implementation.

Example

- STRUCT

Note

- Shows field-level structure for the current area
- Non-mutating inspection command
- STRUCT with no arguments reports field and index information for the current area.
- STRUCT INDEX is explicit index-info mode; index info is included by default.
- STRUCT FIELDS suppresses index info and reports fields only.
- STRUCT ALL reports all open areas.
- STRUCT ALL VERBOSE includes verbose CNX tag information where available.
- STRUCT is read-only; it reports structure/index metadata and does not mutate table data.
- Keeping this file minimal avoids duplicate cmd_STRUCT definitions.
- Future shared STRUCT helpers may live here without adding command dispatch.

Provenance

- Topic key: DOT|STRUCT
- Included HELP rows: 26
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STU_REPEAT

STU_REPEAT

- Catalog/topic: DOT / STU_REPEAT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Student/demo repeat command used to teach argument handling, loops, and command output.

Status

- implemented=yes; supported=yes

Syntax

- STU_REPEAT [USAGE|<text...>]

Provenance

- Topic key: DOT|STU_REPEAT
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STU_UPPER

STU_UPPER

- Catalog/topic: DOT / STU_UPPER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Student/demo uppercase command used to teach string handling and command output.

Status

- implemented=yes; supported=yes

Syntax

- STU_UPPER [USAGE|<text...>]

Provenance

- Topic key: DOT|STU_UPPER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STUDENTECHO

STUDENTECHO

- Catalog/topic: DOT / STUDENTECHO
- Status: supported
- Implemented/supported: T / T

- Primary/confidence: DOTREF / CATALOG

Summary Student/demo echo command used to teach command registration, argument handling, and output.

Status

- implemented=yes; supported=yes

Syntax

- STUDENTECHO [USAGE|<text...>]

Provenance

- Topic key: DOT|STUDENTECHO
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STUDENTHELLO

STUDENTHELLO

- Catalog/topic: DOT / STUDENTHELLO
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Student/demo hello command used to teach the simplest command registration and output path.

Status

- implemented=yes; supported=yes

Syntax

- STUDENTHELLO USAGE

Provenance

- Topic key: DOT|STUDENTHELLO
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

STUFF

STUFF

- Catalog/topic: FOX / STUFF
- Status: supported
- Implemented/supported: F / T
- Primary/confidence: FOXREF / CATALOG

Summary Replace <nLen> characters of <cExpr> starting at <nStart> with <cRepl>.

Status

- implemented=no; supported=yes

Syntax

- STUFF(<cExpr>, <nStart>, <nLen>, <cRepl>)

Provenance

- Topic key: FOX|STUFF
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

SUBSTR

SUBSTR

- Catalog/topic: FOX / SUBSTR
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Return substring of <cExpr> starting at <nStart> for <nLen> characters (or to end).

Status

- implemented=yes; supported=yes

Syntax

- SUBSTR(<cExpr>, <nStart>[, <nLen>])

Provenance

- Topic key: FOX|SUBSTR
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TABLE_BUFFER

TABLE_BUFFER

- Catalog/topic: DOT / TABLE_BUFFER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic helper surface for inspecting or testing TABLE buffering and stale-field behavior.

- Inspect or change per-area table-buffer state.

Status

- implemented=yes; supported=yes

Syntax

- `TABLE_BUFFER [USAGE|<args...>]`

Usage

- `TABLE_BUFFER`
- `TABLE_BUFFER USAGE`
- `TABLE_BUFFER STATUS [ALL]`
- `TABLE_BUFFER BUFFER ON|OFF|DIRTY|CLEAN|STALE|FRESH|STATUS|DUMP|TESTADD|RESET`

Note

- No arguments reports current buffer state. State-changing subcommands mutate session buffer metadata.

Provenance

- Topic key: `DOT|TABLE_BUFFER`
- Included HELP rows: 9
- HELP reference run: `MANRUN-20260717T222026Z-28C704E0`
- Disposition run: `MANRUN-20260717T230554Z-DB3F2DC8`
- Authority: `candidate_only; publication_authority_claimed=0`

TEXT

TEXT

- Catalog/topic: `DOT / TEXT`
- Status: `supported`
- Implemented/supported: `T / T`
- Primary/confidence: `DOTREF / CATALOG`

Summary Text-oriented helper command surface for demonstration, teaching, or local text workflows.

Status

- `implemented=yes; supported=yes`

Syntax

- `TEXT [USAGE|<args...>]`

Provenance

- Topic key: `DOT|TEXT`
- Included HELP rows: 3
- HELP reference run: `MANRUN-20260717T222026Z-28C704E0`
- Disposition run: `MANRUN-20260717T230554Z-DB3F2DC8`
- Authority: `candidate_only; publication_authority_claimed=0`

TIME

TIME

- Catalog/topic: `FOX / TIME`
- Status: `supported`
- Implemented/supported: `T / T`
- Primary/confidence: `FOXREF / CATALOG`

Summary Return the current system time as a character string (HH:MM:SS).

Status

- implemented=yes; supported=yes

Syntax

- TIME()

Provenance

- Topic key: FOX|TIME
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TOP

TOP

- Catalog/topic: DOT / TOP
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Move to the first record in the current table/order.

- Move the current work-area cursor to the first logical record
- Move the current work-area cursor to the first visible/logical record using the shared filter-aware navigation selector.

Status

- implemented=yes; supported=yes

Syntax

- TOP

Usage

- TOP
- TOP USAGE

Example

- TOP

Note

- Uses the active order when an order is active
- Equivalent user intent to GO TOP / GO FIRST
- TOP with no arguments moves to the first visible record.
- TOP requires an open table except for TOP USAGE.
- TOP mutates cursor position but does not mutate table data.
- TALK ON prints the resulting record number.

Provenance

- Topic key: DOT|TOP
- Included HELP rows: 14
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TRIM

TRIM

- Catalog/topic: FOX / TRIM
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Remove trailing spaces from <cExpression> (alias/compat; see RTRIM).

Status

- implemented=yes; supported=yes

Syntax

- TRIM(<cExpression>)

Provenance

- Topic key: FOX|TRIM
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TUPEXPORT

TUPEXPORT

- Catalog/topic: DOT / TUPEXPORT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Export tuple/projection output through the tuple export helper surface.

- Export tuple graph rows to a CSV file using a tuple spec, optional field list, and optional FOR predicate.

Status

- implemented=yes; supported=yes

Syntax

- TUPEXPORT [USAGE|<args...>]

Usage

- TUPEXPORT USAGE
- TUPEXPORT CSV <path>
- TUPEXPORT CSV <path> <tuple-spec>
- TUPEXPORT CSV <path> FIELDS <field-list>
- TUPEXPORT CSV <path> * FOR <expr>

Example

- TUPEXPORT CSV tmpstudents.csv
- TUPEXPORT CSV tmpstudents_names.csv FIELDS LNAME,FNAME
- TUPEXPORT CSV tmpstudents_major.csv STUDENTS.,*MAJORS*. FOR MAJORS.NAME = “CS”

Note

- TUPEXPORT USAGE prints usage before open-table checks or file writes.
- TUPEXPORT writes/truncates the requested CSV file.
- Tuple cursor/workspace cursor state is restored best-effort.

Provenance

- Topic key: DOT|TUPEXPORT
- Included HELP rows: 15
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TUPLE

TUPLE

- Catalog/topic: DOT / TUPLE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Build one tuple row from fields across work areas. Examples: TUPLE * (all fields from current area)
TUPLE LNAME,FNAME (selected fields from

- Build and print a tuple row from the canonical tuple builder using a tuple field specification and optional output flags.
- Build one tuple row from fields across work areas.
- Examples:
- TUPLE * (all fields from current area)
- TUPLE LNAME,FNAME (selected fields from current area)
- TUPLE #11.* (all fields from area 11)
- TUPLE #9.*,#11.LNAME (mix of areas)
- Notes:
- #n prefixes a work area slot; "*" means all fields for that area.
- Output fields are separated by ASCII Unit Separator (0x1F).

Status

- implemented=yes; supported=yes

Syntax

- TUPLE <spec>

Usage

- TUPLE
- TUPLE USAGE
- TUPLE <spec>
- TUPLE <spec> -HEADER
- TUPLE <spec> -AREA-PREFIX
- TUPLE <spec> -NO-ECHO
- TUPLE <spec> -STRICT
- TUPLE <spec> -HEADER-ONLY
- TUPLE <spec> -VALUES-ONLY
- TUPLE <spec> DEBUG
- TUPLE <spec> -DEBUG
- TUPLE <spec> -NULL <token>

Example

- TUPLE
- TUPLE LNAME,FNAME
- TUPLE STUDENTS.LNAME,STUDENTS.FNAME
- TUPLE * -HEADER
- TUPLE * -VALUES-ONLY

Note

- TUPLE with no arguments uses the default star spec.
- TUPLE delegates tuple truth to tuple_builder.
- TUPLE can print formatted output, raw unit-separated values, or both.
- -HEADER prints a header row before values.
- -AREA-PREFIX prefixes header columns with area context.
- -NO-ECHO preserves legacy raw-only scripting behavior.
- -VALUES-ONLY prints raw unit-separated values only.
- DEBUG and -DEBUG print the raw unit-separated row before formatted output.
- -STRICT asks the tuple builder to reject loose field matches.
- TUPLE is read-only for table data.

Provenance

- Topic key: DOT|TUPLE
- Included HELP rows: 39
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TUPLEDELTA

TUPLEDELTA

- Catalog/topic: DOT / TUPLEDELTA
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Tuple diagnostic helper surface for comparing projected tuple output or tuple-state deltas.

- Compare two named tuple streams and report insert, delete, and update deltas. The loader remains a skeleton until tuple stream storage is finalized.

Status

- implemented=yes; supported=yes

Syntax

- TUPLEDELTA [USAGE|<args...>]

Usage

- TUPLEDELTA USAGE
- TUPLEDELTA <baseline-stream> <current-stream>

Note

- TUPLEDELTA requires two stream names except for TUPLEDELTA USAGE.
- Tuple stream loading is not implemented in this skeleton.
- REC_ID or PRIMARY UNIQUE is intended to be the identity key.
- Field-level diffing is intentionally stubbed for now.
- TUPLEDELTA is diagnostic and does not mutate table or index data.

Provenance

- Topic key: DOT|TUPLEDELTA
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TUPTALK

TUPTALK

- Catalog/topic: DOT / TUPTALK
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary DotTalk++ tuple/logical-row command.

- Tuple-based normalization test harness and live DBF capture tool for building, normalizing, dumping, exporting, and pushing tuple entries.

Status

- implemented=yes; supported=yes

Syntax

- TUPTALK

Usage

- TUPTALK
- TUPTALK USAGE
- TUPTALK RESET
- TUPTALK ADD <type> <len> <raw>
- TUPTALK ADD <type> <len> <dec> <raw>
- TUPTALK LIST
- TUPTALK NORMALIZE
- TUPTALK DUMP

- TUPTALK EXPORT CSV
- TUPTALK EXPORT TSV
- TUPTALK EXPORT CSV <path>
- TUPTALK EXPORT TSV <path>
- TUPTALK PUSH <field>
- TUPTALK PUSH ALL
- TUPTALK PUSH ALL FILTER <mask>
- TUPTALK PUSH FILTER <mask>
- TUPTALK PUSH ROW

Example

- TUPTALK PUSH LASTNAME
- TUPTALK PUSH 3
- TUPTALK PUSH ALL
- TUPTALK PUSH ALL FILTER CND
- TUPTALK PUSH ROW

Note

- TUPTALK with no arguments shows usage/help and buffer status.
- TT is wired as a shell alias for TUPTALK.
- ADD creates a test entry from raw text and schema type metadata.
- NORMALIZE fills normalized values using value_normalize.
- PUSH captures fields or rows from the current DBF record.
- EXPORT writes CSV or TSV when a path is supplied or uses default behavior.
- TUPTALK mutates its process-local scratch buffer and may write export files.

Provenance

- Topic key: DOT|TUPTALK
- Included HELP rows: 33
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TUPVALIDATE

TUPVALIDATE

- Catalog/topic: DOT / TUPVALIDATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Tuple validation helper surface for checking tuple projection, relation-walk, or logical-row consistency.

- Validate tuple graph rows for the current table using the tuple graph cursor and relation-aware tuple validation layer.

Status

- implemented=yes; supported=yes

Syntax

- TUPVALIDATE [USAGE|<args...>]

Usage

- TUPVALIDATE
- TUPVALIDATE USAGE
- TUPVALIDATE *
- TUPVALIDATE <tuple-spec>
- TUPVALIDATE * FOR <expr>
- TUPVALIDATE * FOR <expr> MAX <n>
- TUPVALIDATE * FOR <expr> TRACE

Example

- TUPVALIDATE LNAME,FNAME
- TUPVALIDATE STUDENTS.,*MAJORS*. FOR MAJORS.NAME = CS

Note

- TUPVALIDATE with no arguments validates the default star tuple spec.
- TUPVALIDATE USAGE prints usage and does not require an open table.
- The tuple graph cursor uses active ordering and relation context.
- Validation checks tuple cells against their source work areas when available.
- Cursor restoration is reported after validation.
- TUPVALIDATE is read-only for table data but moves cursors during validation.

Provenance

- Topic key: DOT|TUPVALIDATE
- Included HELP rows: 19
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TURBOPACK

TURBOPACK

- Catalog/topic: DOT / TURBOPACK
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Turbo Vision / pack-related utility.

- Fast byte-oriented compaction for plain non-memo, non-x64 DBF tables.

Status

- implemented=yes; supported=yes

Syntax

- TURBOPACK

Usage

- TURBOPACK USAGE
- TURBOPACK

Example

- TURBOPACK

Note

- TURBOPACK USAGE prints usage before open-table checks.
- TURBOPACK is a fast path for plain DBF tables only.
- Memo tables and x64 tables are refused; use PACK instead.
- TURBOPACK closes the table on success.
- Index containers must be rebuilt/rebound after TURBOPACK.

Provenance

- Topic key: DOT|TURBOPACK
- Included HELP rows: 12
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

TVISION

TVISION

- Catalog/topic: DOT / TVISION
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Turbo Vision diagnostics / demos.

- Confirm Turbo Vision/TUI support is linked; this is currently a stub.

Status

- implemented=yes; supported=yes

Syntax

- TVISION

Usage

- TVISION
- TVISION USAGE

Example

- TVISION

Note

- TVISION with no arguments prints the current TUI support status.
- TVISION USAGE prints usage.
- This command does not launch a TUI yet.

Provenance

- Topic key: DOT|TVISION
- Included HELP rows: 10
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

UNDELETE

UNDELETE

- Catalog/topic: DOT / UNDELETE
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Compatibility alias for RECALL to unmark the current deleted record.

Status

- implemented=yes; supported=yes

Syntax

- UNDELETE

Provenance

- Topic key: DOT|UNDELETE
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: `candidate_only`; `publication_authority_claimed=0`

UNLOCK

UNLOCK

- Catalog/topic: DOT / UNLOCK
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Release a previously acquired lock (optionally all locks).

- Release the current record lock, a specified record lock, or the table lock.

Status

- implemented=yes; supported=yes

Syntax

- UNLOCK [ALL]

Usage

- UNLOCK USAGE
- UNLOCK
- UNLOCK <recno>
- UNLOCK ALL
- UNLOCK TABLE

Example

- UNLOCK
- UNLOCK 10
- UNLOCK ALL
- UNLOCK TABLE

Note

- UNLOCK USAGE returns before open-table checks.
- UNLOCK with no arguments unlocks the current record.
- UNLOCK ALL and UNLOCK TABLE release the table lock.

Provenance

- Topic key: DOT|UNLOCK
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

UNTIL

UNTIL

- Catalog/topic: DOT / UNTIL
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Begin an UNTIL block (scripting).

- Buffer and execute an UNTIL...ENDUNTIL loop from the current record until a boolean expression becomes true.

Status

- implemented=yes; supported=yes

Syntax

- UNTIL <expr>

Usage

- UNTIL USAGE
- UNTIL <bool-expr> [QUIET]
- ENDUNTIL
- ENDUNTIL USAGE

Example

- UNTIL EOF()
- TUPLE LNAME,FNAME,GPA
- ENDUNTIL

Note

- UNTIL USAGE and ENDUNTIL USAGE do not start or execute a loop.
- UNTIL starts buffering; the shell must route body lines to UNTIL_BUFFER.
- ENDUNTIL executes buffered body lines through the canonical loop executor.
- Execution starts at the current record and advances one record per iteration.
- Buffered body command effects are owned by those commands.

Provenance

- Topic key: DOT|UNTIL
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

UNTIL_BUFFER

UNTIL_BUFFER

- Catalog/topic: DOT / UNTIL_BUFFER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic helper surface for inspecting or testing buffered UNTIL control-flow behavior.

Status

- implemented=yes; supported=yes

Syntax

- UNTIL_BUFFER [USAGE|<args...>]

Provenance

- Topic key: DOT|UNTIL_BUFFER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

UPDATE

UPDATE

- Catalog/topic: DOT / UPDATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Update data rows (scripting/SQL helper; see command usage).

- Update records in the current DBF work area using SQL-like SET/WHERE syntax.

Status

- implemented=yes; supported=yes

Syntax

- UPDATE <statement>

Usage

- UPDATE USAGE
- UPDATE SET <field>=<value>[, ...] [WHERE <expr>]

Example

- UPDATE SET GPA=3.5 WHERE SID = 1001
- UPDATE SET MAJOR="CSCI" WHERE MAJOR = "CS"

Note

- UPDATE USAGE prints usage before open-table checks.
- UPDATE without WHERE may update all visible records depending on implementation.
- Use WHERE intentionally.

Provenance

- Topic key: DOT|UPDATE
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

UPPER

UPPER

- Catalog/topic: FOX / UPPER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: FOXREF / CATALOG

Summary Convert <cExpression> to upper-case.

Status

- implemented=yes; supported=yes

Syntax

- UPPER(<cExpression>)

Provenance

- Topic key: FOX|UPPER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

USE

USE

- Catalog/topic: DOT / USE
- Status: **supported**
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Open a DBF table in the active DotTalk++ work area.

- Open a DBF table into the current work area, with duplicate-open guard, memo auto-attach, optional index auto-attach, and NOINDEX physical-order mode.

Status

- implemented=yes; supported=yes

Syntax

- USE <table> [ALIAS <name>] [NOINDEX]

Usage

- USE USAGE
- USE <table>
- USE <table.dbf>
- USE <pathtable.dbf>
- USE <table> NOINDEX
- USE <table> NOIDX

Note

- USE requires a table name or path; no usable argument shows usage.
- Relative logical names resolve through the configured DBF path slot.
- USE prevents duplicate opens of the same DBF path across work areas.
- USE clears stale order/tag/container state and closes the current area before opening the new DBF.
- USE opens the target DBF and populates DbArea metadata.
- USE auto-attaches memo storage when memo fields are present.
- USE auto-attaches flavor-appropriate indexes when present, unless NOINDEX/NOIDX is specified.
- USE prefers the configured INDEXES slot and falls back to the DBF directory.
- NOINDEX/NOIDX opens the table in physical order and skips index auto-attach.
- USE is a session/area mutation command; it changes the current work area binding but should not mutate table records.

Provenance

- Topic key: DOT|USE
- Included HELP rows: 20
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

VAL

VAL

- Catalog/topic: FOX / VAL
- Status: **supported**
- Implemented/supported: T / T

- Primary/confidence: FOXREF / CATALOG

Summary Convert a numeric-looking character expression to a number.

Status

- implemented=yes; supported=yes

Syntax

- VAL(<cExpression>)

Provenance

- Topic key: FOX|VAL
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

VALIDATE

VALIDATE

- Catalog/topic: DOT / VALIDATE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Schema/sidecar validation command.

- Route validation subcommands such as VALIDATE UNIQUE to their handlers.

Status

- implemented=yes; supported=yes

Syntax

- VALIDATE <path>

Usage

- VALIDATE USAGE
- VALIDATE UNIQUE USAGE
- VALIDATE UNIQUE FIELD <name> [IGNORE DELETED] [REPAIR] [REPORT TO <path>]

Example

- VALIDATE UNIQUE FIELD SID
- VALIDATE UNIQUE FIELD EMAIL IGNORE DELETED
- VALIDATE UNIQUE FIELD SID REPAIR
- VALIDATE UNIQUE FIELD SID REPORT TO tmpsid_dupes.txt

Note

- VALIDATE with no arguments prints usage.
- VALIDATE USAGE prints usage and does not scan or repair records.
- VALIDATE UNIQUE is delegated to the UNIQUE validator.
- REPAIR may mutate field values; use it intentionally.

Provenance

- Topic key: DOT|VALIDATE
- Included HELP rows: 15
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

WEB

WEB

- Catalog/topic: DOT / WEB
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Web-oriented helper command surface for local documentation, preview, or integration workflows where enabled.

- Open, fetch, or inspect web URLs using the default handler or WinHTTP.

Status

- implemented=yes; supported=yes

Syntax

- WEB [USAGE]<args...>

Usage

- WEB USAGE
- WEB OPEN <url>
- WEB LAUNCH <url>
- WEB GET <url>
- WEB HEAD <url>
- WEB FETCH <url> TO <file>

Note

- WEB USAGE prints usage and does not launch a browser, make a network request, or write files.
- WEB OPEN/LAUNCH use the OS default URL handler.
- WEB GET/HEAD use HTTP request support where implemented.
- WEB FETCH writes the response body to the requested file.

Provenance

- Topic key: DOT|WEB
- Included HELP rows: 14
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

WHILE

WHILE

- Catalog/topic: DOT / WHILE
- Status: supported
- Implemented/supported: T / T

- Primary/confidence: DOTREF / CATALOG

Summary Begin a WHILE block (scripting).

- Buffer and execute a WHILE...ENDWHILE loop from the current record while a boolean expression remains true.

Status

- implemented=yes; supported=yes

Syntax

- WHILE <expr>

Usage

- WHILE USAGE
- WHILE <bool-expr> [QUIET]
- ENDWHILE
- ENDWHILE USAGE

Example

- WHILE GPA >= 3.0
- TUPLE LNAME,FNAME,GPA
- ENDWHILE

Note

- WHILE USAGE and ENDWHILE USAGE do not start or execute a loop.
- WHILE starts buffering; the shell must route body lines to WHILE_BUFFER.
- ENDWHILE executes buffered body lines through the canonical loop executor.
- Execution starts at the current record and advances one record per iteration.
- Buffered body command effects are owned by those commands.

Provenance

- Topic key: DOT|WHILE
- Included HELP rows: 16
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

WHILE_BUFFER

WHILE_BUFFER

- Catalog/topic: DOT / WHILE_BUFFER
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Developer/diagnostic helper surface for inspecting or testing buffered WHILE control-flow behavior.

Status

- implemented=yes; supported=yes

Syntax

- WHILE_BUFFER [USAGE|<args...>]

Provenance

- Topic key: DOT|WHILE_BUFFER
- Included HELP rows: 3
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

WORKSPACE

WORKSPACE

- Catalog/topic: DOT / WORKSPACE
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Canonical live work-area/session command. Usage: WORKSPACE List current open work areas.
WORKSPACE CLOSE Close all work areas and clear

- Manage live work-area/session state
- Report and manage live work-area/session layout.
- Canonical live work-area/session command.
- Usage:
- WORKSPACE
- List current open work areas.
- WORKSPACE CLOSE
- Close all work areas and clear relation state.
- WORKSPACE OPEN DBF
- Scan the configured DBF path slot and open tables into areas 0..N.
- WORKSPACE OPEN <dir>
- Scan a specific directory and open DBFs into areas 0..N.
- WORKSPACE SAVE <name>
- Save a workspace layout where supported.
- WORKSPACE LOAD <name>
- Restore a saved workspace layout where supported.
- Notes:
- WORKSPACE owns live areas, aliases, index/tag bindings, and relation/session layout.
- DDL owns schema/definition work.
- WSREPORT owns verbose diagnostics.

Status

- implemented=yes; supported=yes

Syntax

- WORKSPACE
- WORKSPACE OPEN DBF
- WORKSPACE OPEN <dir>
- WORKSPACE ADD <file.dbf>
- WORKSPACE CLOSE
- WORKSPACE SAVE <name>
- WORKSPACE LOAD <name>
- WORKSPACE [OPEN <DBF|dir>|CLOSE|SAVE <name>|LOAD <name>]

Usage

- WORKSPACE
- WORKSPACE USAGE
- WORKSPACE ALL
- WORKSPACE OPEN DBF
- WORKSPACE OPEN <dir>
- WORKSPACE OPEN <file.dbf>
- WORKSPACE ADD <file.dbf>
- WORKSPACE ADD <target> CNX [FALLBACK] [TABLE]
- WORKSPACE ADD <target> INX|IDX [FALLBACK] [TABLE]
- WORKSPACE ADD <target> CDX [FALLBACK] [TABLE]
- WORKSPACE OPEN <target> CNX [FALLBACK] [recursive] [TABLE]
- WORKSPACE OPEN <target> INX|IDX [FALLBACK] [recursive] [TABLE]
- WORKSPACE OPEN <target> CDX [FALLBACK] [recursive] [TABLE]
- WORKSPACE CLOSE
- WORKSPACE CLOSE <n> [m ...]
- WORKSPACE CLOSE <name|file|stem|alias>[,...]
- WORKSPACE SAVE <file>
- WORKSPACE LOAD <file>
- WORKSPACE TUPLES [LIMIT <n>] [OFFSET <n>] [AREA <n>]

Example

- WORKSPACE
- WORKSPACE CLOSE
- WORKSPACE OPEN DBF
- WORKSPACE ADD students
- WORKSPACE SAVE mcc
- WORKSPACE LOAD mcc.dtschemas

Note

- WORKSPACE owns live areas, aliases, orders, and relation/session layout
- WORKSPACE OPEN replaces area membership; WORKSPACE ADD preserves existing areas
- DDL owns schema/definition work
- SCHEMAS remains a compatibility shim for older scripts

Provenance

- Topic key: DOT|WORKSPACE
- Included HELP rows: 58
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

WSREPORT

WSREPORT

- Catalog/topic: DOT / WSREPORT
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary

Workspace report / diagnostics.

- Print a workspace/status report covering open areas, LMDB/order summary, and table-buffer state.

Status

- implemented=yes; supported=yes

Syntax

- WSREPORT

Usage

- WSREPORT
- WSREPORT USAGE
- WSREPORT ALL

Note

- WSREPORT with no arguments reports the current workspace and current area.
- WSREPORT ALL includes all open work areas in the area/index summary.
- WSREPORT USAGE prints usage and does not inspect areas.
- WSREPORT is read-only.

Provenance

- Topic key: DOT|WSREPORT
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ZAP

ZAP

- Catalog/topic: DOT / ZAP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Delete all records from the current table.

- Remove all records from the current non-memo DBF while preserving structure.

Status

- implemented=yes; supported=yes

Syntax

- ZAP

Usage

- ZAP USAGE
- ZAP

Example

- ZAP

Note

- ZIP USAGE prints usage before open-table checks.
- ZIP rewrites the current DBF with zero records and closes the table on success.
- ZIP currently refuses memo tables.
- Index containers must be rebuilt/rebound afterward.

Provenance

- Topic key: DOT|ZIP
- Included HELP rows: 11
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

ZIP

ZIP

- Catalog/topic: DOT / ZIP
- Status: supported
- Implemented/supported: T / T
- Primary/confidence: DOTREF / CATALOG

Summary Archive helper command surface for ZIP-oriented local packaging or inspection workflows.

- List, create, or extract ZIP archives through the configured ZIP backend.

Status

- implemented=yes; supported=yes

Syntax

- ZIP [USAGE|<args...>]

Usage

- ZIP USAGE
- ZIP LIST <archive.zip>
- ZIP CREATE <archive.zip> <path>
- ZIP EXTRACT <archive.zip> [target_dir]

Example

- ZIP LIST backups.zip
- ZIP CREATE source_bundle.zip src
- ZIP EXTRACT source_bundle.zip tmpsource_bundle

Note

- ZIP USAGE prints usage and does not touch archive files.
- ZIP LIST reads an archive and prints entries.
- ZIP CREATE writes an archive, adding .zip when needed.
- ZIP EXTRACT writes files under the target directory or current directory.

Provenance

- Topic key: DOT|ZIP
- Included HELP rows: 15
- HELP reference run: MANRUN-20260717T222026Z-28C704E0
- Disposition run: MANRUN-20260717T230554Z-DB3F2DC8
- Authority: candidate_only; publication_authority_claimed=0

Function Reference

63 core functions harvested from `function_catalog.cpp` (FunctionDoc table). Student/extension functions self-register from `src/ext/fn` and are not enumerated here.

Function	Category	Args	Summary
ABS	Numeric	1..1	Return the absolute value of a numeric expression.
ACOS	Numeric	1..1	Return the arc-cosine of a numeric value.
ALLTRIM	String	1..1	Trim leading and trailing spaces from a string value.
ASC	String	1..1	Return the character code of the first character in a string.
ASIN	Numeric	1..1	Return the arc-sine of a numeric value.
AT	Search	2..3	Return the 1-based position of a substring occurrence.
ATAN	Numeric	1..1	Return the arc-tangent of a numeric value.
ATC	Search	2..3	Return the 1-based position of a substring occurrence, case-insensitive.
BETWEEN	Numeric	3..3	Return whether a numeric value lies within an inclusive lower and upper bound.
CDOW	Date	1..1	Return the day-of-week name for a date.
CEILING	Numeric	1..1	Return the smallest integer greater than or equal to the numeric value.
CHR	String	1..1	Return the single-character string for a numeric character code.
CHRTRAN	String	3..3	Replace characters in a string by position-mapped translation.
CMONTH	Date	1..1	Return the month name for a date.

Function	Category	Args	Summary
CONCAT	Construction	1..32	Concatenate all arguments into a single string.
COS	Numeric	1..1	Return the cosine of an angle expressed in radians.
CTOD	Date	1..1	Convert a character string to a date value.
DATE	Date	0..0	Return the current system date.
DATEADD	Date	2..2	Return a new date shifted by a specified interval.
DATEDIFF	Date	2..2	Return the difference between two date values.
DATETIME	Date	0..0	Return the current date/time value.
DAY	Date	1..1	Return the day-of-month component of a date.
DOW	Date	1..1	Return the day-of-week number for a date.
DTOC	Date	1..2	Convert a date value to character form.
DTOS	Date	1..1	Convert a date value to sortable character form.
EMPTY	Logical	1..1	Return whether a value is empty according to expression-engine rules.
EXP	Numeric	1..1	Return e raised to the power of the supplied numeric value.
FLOOR	Numeric	1..1	Return the largest integer less than or equal to the numeric value.
GOMONTH	Date	2..2	Return a date shifted by a specified number of months.
INT	Numeric	1..1	Return the integer portion of a numeric value.
LEFT	String	2..2	Return the leftmost characters of a string.
LEN	Numeric	1..1	Return the length of a string value.
LIKE	Logical	2..2	Match a value against a wildcard pattern.
LOG	Numeric	1..1	Return the natural logarithm of a numeric value.
LOG10	Numeric	1..1	Return the base-10 logarithm of a numeric value.
LOWER	String	1..1	Convert text to lowercase.

Function	Category	Args	Summary
LTRIM	String	1..1	Trim leading spaces from a string value.
MAX	Numeric	2..32	Return the largest value from two or more numeric expressions.
MIN	Numeric	2..32	Return the smallest value from two or more numeric expressions.
MOD	Numeric	2..2	Return the remainder from dividing one numeric value by another.
MONTH	Date	1..1	Return the month component of a date.
NOW	Date	0..0	Return the current date/time value.
RAND	Numeric	0..0	Return a pseudo-random numeric value.
RAT	Search	2..2	Return the 1-based position of the last occurrence of a substring.
REPLICATE	Construction	2..2	Repeat a string value a specified number of times.
RIGHT	String	2..2	Return the rightmost characters of a string.
ROUND	Numeric	1..2	Round a numeric value to a specified number of decimal places.
RTRIM	String	1..1	Trim trailing spaces from a string value.
SECONDS	Date	0..0	Return the current time expressed as seconds-of-day or equivalent engine-specific seconds value.
SIN	Numeric	1..1	Return the sine of an angle expressed in radians.
SOUNDEX	Search	1..1	Return a four-character phonetic Soundex code for a character expression.
SPACE	Construction	1..1	Return a string containing the requested number of spaces.
SQRT	Numeric	1..1	Return the square root of a numeric value.
STR	Conversion	1..3	Convert a numeric value to a formatted string.
STRTRAN	String	3..3	Replace occurrences of a substring within a string.

Function	Category	Args	Summary
SUBSTR	String	2..3	Return a substring starting at a 1-based position.
TAN	Numeric	1..1	Return the tangent of an angle expressed in radians.
TIME	Date	0..0	Return the current system time.
TODAY	Date	0..0	Return the current system date.
TRANSFORM	Conversion	1..2	Convert a value to display-oriented character form.
UPPER	String	1..1	Convert text to uppercase.
VAL	Conversion	1..1	Extract the leading numeric portion of a string.
YEAR	Date	1..1	Return the year component of a date.

Appendix: Review and Deferred: SET-family

Status: SECTION_SKELETON_DRAFT_REPAIRED / REVIEW_REQUIRED

Purpose: Skeleton section generated from the revised manual TOC draft.

Promotion boundary: - This section is a structural draft. - Linked command pages are evidence-backed drafts, not final prose. - Review before promotion into the finished Developer Manual.

Commands in this section

- SET - deferred
- SET CASE - deferred
- SET FILTER - deferred
- SET INDEX - deferred
- SET ORDER - deferred
- SET PATH - deferred
- SET RELATION - deferred
- SET UNIQUE - deferred
- SET VAR - aliases: SET VAR! - deferred
- SETCASE - deferred
- SETCDX - deferred
- SETCNX - deferred
- SETFILTER - deferred
- SETINDEX - deferred
- SETLMDB - deferred
- SETNEAR - deferred
- SETORDER - deferred
- SETPATH - deferred

Notes for future prose pass

- Add narrative explanation for this section after command-page sampling.
- Keep runtime-proven evidence separate from design intent.
- Preserve command/help/catalog provenance when promoting prose.

Error / Message Catalog

Canonical error codes use an HRESULT-style 32-bit packing (severity · facility · code) defined in `src/cli/xbase_error_codes.cpp`.

Severities: success=0, warning=1, error=2.

Facilities (subsystems):

Facility	Value
<code>general</code>	0x0001
<code>dbf64</code>	0x0002
<code>fpt64</code>	0x0003
<code>security</code>	0x0004
<code>cli</code>	0x0005
<code>io</code>	0x0006
<code>runtime</code>	0x0007

Review candidate. The full code→message enumeration is sourced from `src/cli/message_catalog.cpp`, `src/cli/help_errors.cpp`, and `src/help/helpdata_messages.cpp`; harvesting the complete message table is a later refinement of `assembler:message-catalog`. The severity/facility taxonomy above is source-derived and final.

Tables, Records, and Data Model

Pippets used: - PIP-004 Reviewed Candidate Pippet - PIP-003 Evidence Review Pippet - PIP-011 Source Reference Inventory Pippet

Evidence lineage: - MDO-120 original draft - MDO-128 source-lane repaired draft - MDO-133 field-note drift repaired v4 draft - MDO-134 evidence review gate

Repair reason: - MDO-121 detected evidence/prose drift in the first Tables, Records, and Data Model draft. - MDO-128 recovers or rebuilds the repair plan, then generates the source-lane repaired draft.

Evidence class: - Repaired draft assembled from MDO-120 draft intent, MDO-121 evidence gate where available, source-lane crosswalk or recovered repair plan, and registry v4. - Runtime behavior remains the source of truth. - This draft does not mutate HELP, META, CMDHELPCHK, catalogs, source files, or production SelfDoc metadata.

Purpose of this section

This section explains the table, record, field, schema, memo, and record-view vocabulary that later manual sections depend on. It follows the Getting Started and Workspaces sections: the reader should already know how the system starts, how areas frame context, and why the manual carries evidence boundaries.

The repaired draft is deliberately conservative. It does not assume every linked evidence page is the same kind of source. Some pages support current DotTalk command prose, some are educational concepts, some are compatibility references, and some need additional review before being described as runtime behavior.

Source-lane rule for this section

A DOTREF/current evidence item may support command-facing prose after evidence review. An EDREF concept item explains vocabulary or teaching structure. A FOXREF compatibility item must be gated before being described as current runtime behavior. SQL or shell reference lanes should be routed to bridge or appendix material.

This rule prevents concept pages such as `TABLE_RECORD_FIELD` from being documented as if they were ordinary runtime commands, and prevents compatibility-only material from being promoted as current DotTalk behavior without proof.

Tables, records, and fields

The core data model starts with tables, records, and fields. TABLE supports table-level vocabulary. RECORD and RECNO support record identity and record-position vocabulary. FIELDS and FIELDMGR support field-level vocabulary and field-management evidence.

TABLE_RECORD_FIELD is concept evidence for the relationship among tables, records, and fields. It should explain the model, not be presented as a user command unless a separate runtime command page proves that behavior.

Schema and DDL

SCHEMA and DDL belong together in this section because they describe structure. SCHEMA supports table-structure vocabulary. DDL supports definition-level or schema-language vocabulary. This draft keeps DDL wording structural and avoids claiming specific DDL behavior until command evidence and runtime examples are reviewed.

Record views and memo fields

RECORDVIEW supports record-oriented presentation or view vocabulary. MEMO supports memo-field discussion. Memo behavior is a sensitive storage and persistence topic, so final prose should distinguish visible memo fields from backend memo storage and should not overclaim persistence details without runtime evidence.

Work areas as context, not the main topic

WORKAREA may appear in the evidence set, but the main explanation of work areas belongs in the already promoted Workspaces section. Here, WORKAREA should be used only to remind the reader that table and record commands operate in a current area/session context.

Compatibility and bridge material

Compatibility evidence appears in this section and must be handled cautiously:

- DBAREAS: appears as compatibility evidence. Do not promote it as current DotTalk behavior without runtime proof.

Compatibility evidence can support historical or compatibility notes, but it should not be promoted as current DotTalk behavior without runtime proof.

Command and concept map

- DBAREA [DOT_COMMAND_EVIDENCE / DOTREF]: is current/DotTalk reference evidence and may support command prose after evidence review.
- DBAREAS [COMPATIBILITY_EVIDENCE / FOXREF]: appears as compatibility evidence. Do not promote it as current DotTalk behavior without runtime proof.
- DDL [DOT_COMMAND_EVIDENCE / DOTREF]: describes schema-definition language or definition-level behavior. Treat it as structural doctrine, not a simple record command.
- FIELDMGR [DOT_COMMAND_EVIDENCE / DOTREF]: belongs to field-management evidence. Use it to discuss how field metadata and field-level operations are organized, not as ordinary user prose without review.
- FIELDS [DOT_COMMAND_EVIDENCE / DOTREF]: supports field-list and field-description discussion.
- MEMO [DOT_COMMAND_EVIDENCE / DOTREF]: supports memo/large-object field discussion. Keep memo storage details conservative unless command evidence or runtime transcripts prove more.
- RECNO [DOT_COMMAND_EVIDENCE / DOTREF]: supports record-position vocabulary.
- RECORD [DOT_COMMAND_EVIDENCE / DOTREF]: supports current-record or record-identity vocabulary.
- RECORDVIEW [DOT_COMMAND_EVIDENCE / DOTREF]: supports record-oriented presentation or viewing vocabulary.
- SCHEMA [DOT_COMMAND_EVIDENCE / DOTREF]: supports table-structure and schema vocabulary.
- TABLE [DOT_COMMAND_EVIDENCE / DOTREF]: supports table-level vocabulary.

- TABLE_RECORD_FIELD [CONCEPT_EVIDENCE / EDREF_CONCEPT]: is concept evidence for the relationship among tables, records, and fields. Do not treat it as a runtime command unless separately proven.
- WORKAREA [CONCEPT_EVIDENCE / EDREF_CONCEPT]: is concept or workspace-adjacent evidence. Mention only as context and keep detailed work-area behavior in the Workspaces section.

What this section should not do yet

- Do not document unrelated commands from the first draft that are not in the linked evidence set.
- Do not treat EDREF concept pages as ordinary runtime commands.
- Do not treat FOXREF compatibility entries as current DotTalk behavior without runtime evidence.
- Do not add examples for mutating or destructive behavior until syntax and runtime transcripts are sampled.
- Do not move detailed workspace/session behavior out of the Workspaces section.

Review notes before candidate generation

- Rerun PIP-003 evidence review on this repaired draft.
- Confirm the drift rows from MDO-121 are cleared or intentionally explained.
- Confirm the command/concept split is acceptable.
- Confirm MEMO wording does not overclaim backend persistence.
- Confirm DDL wording stays structural until examples are proven.

Boundary

- repaired prose draft only
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Command Surface, Dispatch, and Entry Variants

Pippets used: - PIP-001 Target Selection - MDO-166 Target Selection - MDO-167 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches parser dispatch, public command surface, internal command-family ownership, aliases, variants, subcommands, function bridge behavior, and generated command-reference canaries. - Do not send this directly to generic PIP-003. - Run a slow-lane command-surface review first.

Evidence tokens under review: - COMMANDS - SYSCMD - SYSSUBCMD - SYSENTVAR - CMDKEY - CAN_NAME - QUAL_NAME - TOKEN - HANDLER - VIS - PUB_SURF - DISP_REACH - HELP - CMDHELPCHK - parser - dispatch - handler - command surface - entry variant - alias - subcommand - canonical command - AGGS - SET family - FUNCTION bridge

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Generated command draft pages remain draft evidence, not final command reference prose. - Public command surface must be separated from internal owner/family scaffolding. - AGGS is treated as internal/family-owner evidence unless explicitly accepted as a public command surface. - SET-family canonicalization remains deferred unless separately repaired or accepted. - Function bridge behavior must preserve scalar/function entry while respecting command ownership. - SYSCMD, SYSSUBCMD, SYSENTVAR, SYSARGS, SYSHELP, SYSMSG, and SYSFUNC remain future/current feeders even when sparse.

Purpose of this section

This section explains how the Developer Manual should talk about command surface, parser dispatch, command handlers, canonical command identity, aliases, entry variants, subcommands, and function-bridge entry.

It follows Navigation, Indexing, Expressions, and Messages because those sections repeatedly depended on command/function routing and ownership boundaries. The manual now needs a section that explains how commands should be described without collapsing public command surface, internal command-family owners, generated command evidence, parser dispatch, and future metadata feeders into one authority.

The goal is not to publish the final command reference. The goal is to preserve a safe model for developer prose: command surface is what the user can intentionally invoke; dispatch is how input reaches implementation; metadata organizes identity and variants; HELP explains; CMDHELPCHK validates; runtime/source evidence still decides behavior and ownership.

Evidence lanes

This draft uses several evidence lanes.

Current command evidence lane: - COMMANDS - generated command pages - HELP command topics - CMDHELPCHK reports - HELP_COMMANDS exports where available

Current metadata evidence lane: - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - SYSMSG - SYSFUNC

Dispatch concept lane: - parser - dispatch - handler - command surface - public command - internal family owner - entry variant - alias - subcommand - canonical command - function bridge

Canary lane: - AGGS internal owner exposure; - SET family canonicalization; - command/function bridge behavior; - generated command duplicate and slug-collision evidence; - sparse metadata feeder coverage; - aliases and variants that should not be collapsed into canonical commands without review.

Authority boundaries

The same doctrine applies here:

- Runtime proves behavior.
- Source defines implementation and subsystem ownership.
- HELP explains command surface and intended usage.
- Metadata organizes canonical identity, variants, arguments, handlers, visibility, and help alignment.
- CMDHELPCHK validates HELP/catalog consistency.
- SelfDoc preserves provenance.
- Manualgen assembles.

The command-surface section must not let generated command pages or metadata rows become stronger proof than they are. A generated command page is draft evidence. A metadata row organizes identity. A HELP row explains. A handler in source defines implementation. A runtime test proves observed behavior.

Command surface

Command surface means the command vocabulary that a user can intentionally invoke.

Safe wording: - A public command surface should be documented as public only when HELP, metadata, source, and/or runtime evidence support that status. - A word that appears in generated command pages is not automatically a final public command. - A word that appears as an internal owner or command family should not be promoted as user-facing without explicit acceptance. - Visibility and public-surface flags should eventually be checked against metadata such as VIS and PUB_SURF.

This matters for command families, scaffolding, debug commands, and internal owner commands.

GENERIC remains a developer-utility canary `GENERIC` is present in the supported DOTREF surface as a developer utility placeholder. Keep it visible for dispatch and metadata reconciliation, but do not present it as a normal user workflow until runtime behavior and intended audience are separately established. Its current value is as a command-surface canary: registration, `HELP` identity, handler ownership, and runtime availability can be compared without inventing stronger behavior prose.

Canonical commands

A canonical command is the main command identity used for documentation, metadata, and command-reference organization.

Safe wording: - A canonical command may have aliases or entry variants. - A canonical command may own subcommands. - A generated page slug is not automatically the canonical identity. - A handler name is implementation evidence, not necessarily the public name. - `CMDKEY`, `CAN_NAME`, and `QUAL_NAME` should eventually help separate command identity levels.

The manual should preserve a difference between: - token typed by a user; - canonical command identity; - qualified command/subcommand identity; - generated page path; - source handler; - metadata row; - `HELP` topic.

Subcommands and command families

Some command surfaces are naturally scoped through families or subcommands.

Examples needing careful treatment: - `SET` family - `SET ORDER` - `SET INDEX` - `REL` variants - `WORKSPACE LOAD` - `CODASYL LOAD` - aggregate-family ownership - `AGGS` as a possible internal owner

Safe wording: - A family or owner command may organize related verbs or subcommands. - A subcommand should be documented under its scoped owner when evidence supports that structure. - A family name should not be treated as public executable command unless accepted and evidenced. - `SET`-family canonicalization remains deferred until repaired or explicitly accepted.

Aliases and entry variants

Aliases and entry variants are not automatically separate canonical commands.

Safe wording: - An alias may route to a canonical command. - An entry variant may preserve compatibility vocabulary, shortcut spelling, or a user-facing convenience form. - A generated command page may represent an alias or variant rather than a canonical command. - `SYSENTVAR` should eventually organize aliases, variants, shortcut spellings, compatibility forms, and app-style entry points.

Manual prose should avoid treating every distinct token as a separate command until dedupe and variant review are complete.

Application-style UI entry points `ARCTICTALK` and `FOXPRO` are application launch entries rather than ordinary data commands. When Turbo Vision support is present, `ARCTICTALK` opens the ArcticTalk shell and `FOXPRO` opens the FoxPro-style workbench. `FOXTALK` is the legacy alias for `ARCTICTALK`.

These names may be registered or documented even when a particular build does not contain the corresponding Turbo Vision surface. Describe availability from the selected build and runtime evidence; do not infer universal availability from registration alone. Neither entry currently exposes a separate usage branch, so the bare command form is the documented launch surface.

Parser dispatch and handlers

Parser dispatch is the route from user input to implementation. A handler is the implementation endpoint or command function that receives the routed input.

Safe wording: - Parser dispatch and handler ownership require source evidence. - Runtime evidence proves observed routing behavior. - `HELP` and metadata can explain or organize dispatch but do not prove runtime routing. - Handler names may be internal and should not be exposed as public command names unless explicitly intended.

This section should avoid claiming the exact parser algorithm unless source/runtime evidence is attached.

Function bridge behavior

DotTalk++ allows some function/app forms to be used directly from the command line. Prior runtime notes include examples such as UPPER and LEFT.

This is a command-surface canary because function names can look like command verbs.

Safe wording: - FUNCTION bridge behavior should preserve scalar/function entry where supported. - A function-style command-line entry does not necessarily make the function a command. - A command verb may shadow a scalar function name. - MIN/MAX ambiguity remains a separate canary from the expression section. - SYSFUNC and SYSENTVAR should eventually help organize function-command bridge forms.

AGGS boundary

AGGS is a known command-surface canary.

Current doctrine: - AGGS is intended as an internal owner or aggregate-family grouping. - Direct aggregate verbs such as SUM, AVG, MIN, and MAX are the intended user-facing aggregate command surface where evidenced. - AGGS appearing as executable or printing usage may be scaffold/debug leakage unless explicitly accepted. - Generated command pages for AGGS are draft evidence, not final public-surface proof.

This section should preserve AGGS internal family owner exposure as a visible canary.

SET family boundary

SET-family canonicalization remains deferred.

Safe wording: - SET ORDER and SET INDEX are scoped command surfaces that need canonicalization review. - SET-family pages may exist as generated draft evidence. - Final command reference organization should not be settled in this section without dedicated SET-family review. - Indexing owns order/tag semantics; this section owns command-surface identity and dispatch cautions.

Generated command pages

Generated command pages are useful evidence, but they are not final command reference.

Known generated-page issues: - duplicates; - aliases; - variants; - slug collisions; - SET-family canonicalization; - internal owner exposure; - command/function ambiguity.

Safe wording: - Generated pages identify draft evidence. - They should be deduped and reviewed before publication. - Command reference generation must not delete or rewrite generated command pages during manual draft assembly. - Generated pages should feed review, not replace review.

HELP and CMDHELPCHK

HELP explains command usage, concepts, and warnings. CMDHELPCHK validates HELP/catalog consistency.

Safe wording: - HELP can explain command surface intent. - CMDHELPCHK can detect gaps and drift. - Neither HELP nor CMDHELPCHK replaces runtime proof. - Neither HELP nor CMDHELPCHK replaces source ownership.

Manual assembly may use HELP/META/CMDHELPCHK-first workflow, but truth authority remains role-separated.

Future META alignment

This section should eventually align with the metadata system.

Expected future feeders: - SYSCMD for canonical command identity, command key, canonical name, qualified name, token, handler, visibility, public surface, display reach, owner, source authority, source file, help topic, active flag, and notes. - SYSSUBCMD for command-family subcommands such as SET ORDER, SET INDEX, REL variants, WORKSPACE variants, and other scoped command surfaces. - SYSENTVAR for aliases, entry variants, shortcut spellings, compatibility forms, app-style function entries, and reviewed variants. - SYSARGS for command argument

shapes, expression arguments, predicates, tag names, scopes, deleted filters, required arguments, and repeatable arguments. - SYSHelp for help text connected to command owners, canonical commands, subcommands, variants, warnings, examples, and reference material. - SYSMSG for parser diagnostics, unknown command messages, syntax errors, invalid arguments, and ambiguity warnings. - SYSFUNC for function-command bridge cases where scalar function entry overlaps command-like syntax. - HELP_COMMANDS and generated command pages as draft evidence lanes that require dedupe, alias, and slug-collision review.

Temporary evidence is acceptable only when marked as temporary and crosswalked to future META feeders.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- AGGS internal family owner exposure
- SET family canonicalization deferred
- command function bridge preserves scalar entry
- generated command pages duplicates aliases slug collisions
- SYSCMD SYSSUBCMD SYSENTVAR sparse feeder alignment
- HELP CMDHELPCHK not runtime source authority
- parser dispatch handler visibility display reach evidence
- aliases variants subcommands canonical commands not collapsed
- public command surface separated from internal scaffolding
- command reference generation no delete rewrite

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- generated command pages are treated as draft evidence;
- public command surface is separated from internal owner/family scaffolding;
- AGGS is not promoted as public command surface without explicit acceptance;
- SET-family canonicalization remains deferred;
- function bridge behavior preserves scalar entry while respecting command ownership;
- generated command pages with duplicates, aliases, and slug collisions remain draft evidence;
- SYSCMD, SYSSUBCMD, and SYSENTVAR are included as feeders even if sparse;
- HELP and CMDHELPCHK are not treated as runtime/source authority;
- parser dispatch, handlers, public visibility, and display reach remain evidence-gated;
- aliases, variants, subcommands, and canonical commands are not collapsed without review;
- command reference generation does not delete or rewrite generated command pages.

Recommended required tokens for later PIP-003: - COMMANDS - SYSCMD - SYSSUBCMD - SYSENTVAR - CMDKEY - CAN_NAME - QUAL_NAME - TOKEN - HANDLER - VIS - PUB_SURF - DISP_REACH - HELP - CMDHELPCHK - parser - dispatch - handler - command surface - entry variant - alias - subcommand - canonical command - AGGS - SET family - FUNCTION bridge

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation

- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Expressions, Querying, and Aggregates

Pippets used: - PIP-001 Target Selection - MDO-151 Target Selection - MDO-152 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches known command/function and parser canaries. - Do not send this directly to generic PIP-003. - Run a slow-lane expression/function/aggregate review first.

Evidence tokens under review: - CALC - CALCWRITE - WHERE - FOR - LOCATE - CONTINUE - SCAN - COUNT - SUM - AVG - MIN - MAX - AGGS - FUNCTION - FUNCTIONS - PREDICATES - MIN() - MAX() - xexpr - DELETED - NOT DELETED - !DELETED

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Generated command draft pages remain draft evidence, not final command prose. - xexpr owns expression evaluation surfaces unless current source proves a narrower owner for a specific path. - SYSFUNC is a future semantic feeder even if current seeding is sparse. - Scalar function forms and aggregate command forms must remain separate until runtime/source evidence closes the ambiguity. - AGGS is treated as internal/family-owner evidence unless explicitly accepted as a user-facing command surface.

Purpose of this section

This section explains expression evaluation, predicates, query filters, aggregate commands, and function surfaces in DotTalk++.

It follows Navigation and Indexing because those sections establish record movement, search context, active order, tags, projection, and relation boundaries. Expressions sit beneath many of those surfaces. Predicates decide which records are considered. Aggregate commands summarize values across records. Function calls support scalar calculation, command-line function applications, and calculated command behavior.

The goal is not to publish a final function reference. The goal is to establish safe developer-manual prose that preserves the boundaries between expression evaluation, command parsing, predicate filtering, aggregate traversal, scalar functions, and future metadata feeders.

Evidence lanes

This draft uses several evidence lanes.

Current DotTalk evidence lane: - CALC - CALCWRITE - WHERE - FOR - LOCATE - CONTINUE - SCAN - COUNT - SUM - AVG - MIN - MAX - AGGS - FUNCTION - FUNCTIONS - PREDICATES - MIN() - MAX() - xexpr - DELETED - NOT DELETED - !DELETED

Generated command-reference lane: - Generated command pages may identify draft command evidence. - They are not final prose and should not be quoted as final command authority. - AGGS and MIN/MAX require extra review because generated pages can expose implementation or family-owner concepts that may not be intended as public command surface.

Runtime evidence lane: - Runtime examples have shown direct aggregate verbs SUM, AVG, MIN, and MAX working against an open table. - Runtime examples have shown WHERE and FOR producing matching aggregate results in tested cases. - Runtime examples have shown deleted-record filters affecting aggregate outputs in tested cases. - Runtime examples have shown command-line function application behavior for functions such as UPPER and LEFT. - Runtime examples have shown MIN/MAX parser ambiguity that must remain canary-sensitive.

Concept lane: - An expression computes a value. - A predicate is an expression used as a true/false condition. - A query filter selects records. - An aggregate command computes a result across records. - A scalar function computes

a result from arguments. - A command parser may route command-like input differently from expression-function input.

Compatibility lane: - xBase/FoxPro lineage can explain vocabulary, but compatibility evidence must not be promoted as current DotTalk behavior without runtime proof. - Function names that overlap with command verbs are especially compatibility-sensitive.

Future META feeder lane: - SYSFUNC should eventually carry canonical function identity, display name, category, argument range, handler, CALC_CALL, PUB_SURF, SELF_REG, MSG_CAT, and active status. - SYSARGS should eventually carry function and command argument shapes, required/repeatable flags, predicate shapes, deleted filters, and expression values. - SYSCMD should eventually carry command identity for COUNT, SUM, AVG, MIN, MAX, CALC, CALCWRITE, LOCATE, CONTINUE, SCAN, and related command surfaces. - SYSSUBCMD should eventually carry aggregate-family or predicate-related subcommands if the command model keeps subcommand ownership. - SYSENTVAR should eventually carry command/function variants, aliases, and app-style function entry points after seed hygiene review. - SYSHELP should eventually carry curated/generated help text for functions, predicates, aggregate commands, and expression concepts. - SYSMMSG should eventually carry expression errors, nonnumeric aggregate values, no-active-table messages, not-found messages, deleted-filter outcomes, and parser ambiguity warnings.

Expression evaluation surfaces

Expression evaluation is shared infrastructure. The manual should treat xexpr as the expression engine unless current source evidence proves otherwise for a specific path.

Expression surfaces include: - CALC - CALCWRITE - command arguments that accept value expressions - predicates in WHERE or FOR style clauses - function calls - calculated values used by aggregate commands

The developer manual should make a distinction between expression syntax and command syntax. A command may accept an expression, but that does not make the command itself the expression engine.

CALC and CALCWRITE

CALC and CALCWRITE are expression-oriented command surfaces. They are useful places to explain evaluated expressions without requiring table traversal.

Safe wording: - CALC evaluates an expression and displays or returns the result according to its command contract. - CALCWRITE evaluates and writes or displays according to its command contract. - Exact output behavior should be verified with HELP and runtime evidence before final wording.

These commands are natural cross-references for function help and the xexpr engine.

Predicates, WHERE, and FOR

A predicate is an expression used as a condition. WHERE and FOR are predicate-bearing clauses or surfaces.

Safe wording: - WHERE and FOR can restrict which records participate in a command where supported. - Runtime evidence has shown matching aggregate results for equivalent WHERE and FOR predicates in tested aggregate cases. - This equivalence should not be generalized across all commands without evidence.

The manual should not imply that WHERE and FOR are always identical. It should say that they can serve related predicate-filter roles and that each command surface must be checked.

LOCATE, CONTINUE, and SCAN

LOCATE, CONTINUE, and SCAN connect predicates with traversal.

Safe distinctions: - LOCATE searches for records matching a condition. - CONTINUE resumes a prior locate-style search where supported. - SCAN iterates over records and may use predicate or scope rules depending on command syntax.

These commands connect this section with Navigation and Indexing: - Navigation owns movement and current record context. - Indexing owns active order and tag-sensitive traversal context. - Expressions own predicate evaluation. - Commands own their own syntax and side effects.

COUNT and aggregate commands

COUNT, SUM, AVG, MIN, and MAX belong to aggregate command discussion when they operate across records.

Safe aggregate wording: - COUNT counts records or matched records according to command syntax and scope. - SUM computes a total for a numeric expression across records. - AVG computes an average for a numeric expression across records. - MIN and MAX compute minimum and maximum aggregate results where command syntax and runtime support them.

The final manual should attach runtime proof for each command surface before claiming exact syntax, deleted-record behavior, or null/empty-set behavior.

Direct aggregate verbs and AGGS

Direct aggregate verbs should be treated as the user-facing command surface where runtime and HELP evidence support them: - SUM - AVG - MIN - MAX

AGGS should be treated carefully. Current doctrine is: - AGGS is intended as an owner/internal grouping for aggregate verbs. - AGGS being executable or printing usage may be debug/scaffold leakage unless explicitly accepted. - Generated command pages for AGGS are draft evidence, not final public-surface proof.

The manual should explain aggregate family ownership without accidentally publishing an internal owner as a user command.

Scalar functions versus aggregate commands

MIN and MAX are the important canary.

There are two concepts: - scalar function form: MIN() or MAX() as a function over supplied arguments; - aggregate command form: MIN or MAX over records.

These must not be collapsed.

Known runtime canary: - command-style input has shown MIN/MAX aggregate behavior. - function-app and command-line function bridging exists for functions such as UPPER and LEFT. - MIN(2,1) versus MIN 2,1 parser behavior must remain canary-sensitive until current runtime/source evidence closes it.

The manual should preserve the ambiguity: - MIN/MAX aggregate commands are command surfaces. - MIN()/MAX() scalar functions are function surfaces if current function registry/runtime evidence confirms them. - Parser dispatch and command/function shadowing must be documented carefully.

Function command-line bridge

DotTalk++ allows some function/app forms to be used directly from the command line.

Examples in prior runtime notes include: - UPPER with a string-like value - LEFT with a value and length argument

This is important because function names may appear command-like when typed at the prompt. The manual should explain that some command-line input can bridge into scalar/function-app evaluation.

However, the bridge must not override command ownership: - A direct command verb may own a word such as MIN or MAX. - A scalar function may also exist with the same name. - The parser/dispatcher decides which surface receives the input.

Deleted-record filters

Aggregate commands may support deleted-record filters.

Evidence-sensitive filter vocabulary: - DELETED - NOT DELETED - !DELETED

Safe wording: - Deleted-record filters may affect which records participate in aggregate results where supported. - Runtime evidence has shown DELETED returning empty/null-like aggregate behavior in tested cases and NOT DELETED matching full nondeleted results in tested cases. - Exact behavior must be verified per command, expression type, and data state.

The manual should not generalize deleted-record behavior beyond tested aggregate surfaces without proof.

Proof boundary: - deleted-record aggregate filters are proof-aware. - DELETED, NOT DELETED, and !DELETED should not be generalized beyond tested aggregate surfaces. - Final wording must remain tied to command, expression type, and data-state evidence.

Error and null behavior

Aggregate and expression commands can produce errors or null-like results.

Examples needing proof-aware wording: - nonnumeric expression supplied to a numeric aggregate; - character expression supplied to AVG; - unknown field or expression; - empty or deleted-only record sets; - no active table; - invalid function argument count.

SYSMSG is the intended future feeder for error symbols, severity, short text, suggested actions, and implementation status.

HELP FUNCTIONS and FUNCTION help

HELP FUNCTIONS and HELP FUNCTION should be included as user-facing discovery surfaces for expression functions.

The manual should distinguish: - command help for command verbs; - function help for expression functions; - generated command draft pages; - future SYSFUNC metadata.

SYSFUNC is important even when sparsely seeded because it is the future semantic feeder for canonical function identity, argument ranges, handlers, public surface status, and message-catalog alignment.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- MIN/MAX scalar function versus aggregate command ambiguity
- AGGS internal family owner exposure
- direct aggregate verbs versus scalar function forms
- command parser function bridge
- WHERE FOR predicate equivalence
- DELETED NOT DELETED !DELETED aggregate filters
- xexpr owns expression evaluation surfaces
- HELP FUNCTIONS FUNCTION name SYSFUNC future feeder
- generated command pages draft evidence
- MIN(2,1) versus MIN 2,1 parser behavior

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Future META alignment

This section should eventually align with the metadata system.

Expected future feeders: - SYSFUNC for canonical function identity, display name, category, argument range, implementation status, visibility tier, owner, source authority, source file, handler, CALC_CALL, PUB_SURF, SELF_REG, MSG_CAT, active status, and notes. - SYSARGS for function and command argument shapes, predicates, deleted filters, repeatable arguments, and required values. - SYSCMD for command identity and handler alignment for CALC, CALCWRITE, COUNT, SUM, AVG, MIN, MAX, LOCATE, CONTINUE, and SCAN. - SYSSUBCMD for aggregate-family or predicate-related subcommands if those are modeled as subcommands. - SYSENTVAR for variants, aliases, and command-line function-app entry points after seed hygiene review. - SYSHELP for generated and curated help text connected to command and function owners. - SYSMSG for expression, aggregate, predicate, and parser diagnostics.

Temporary evidence is acceptable only when marked as temporary and crosswalked to future META feeders.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- generated command pages are treated as draft evidence;
- AGGS is not promoted as a public user command without explicit acceptance;
- MIN/MAX scalar function and aggregate command forms are separated;
- function-command bridge behavior is described conservatively;
- WHERE and FOR equivalence is evidence-gated;
- deleted-record filters are proof-aware;
- xexpr ownership is preserved;
- HELP FUNCTIONS, HELP FUNCTION , and SYSFUNC future feeder notes are present;
- compatibility evidence is not presented as runtime proof;
- parser ambiguity remains visible.

Recommended required tokens for later PIP-003: - CALC - CALCWRITE - WHERE - FOR - LOCATE - CONTINUE - SCAN - COUNT - SUM - AVG - MIN - MAX - AGGS - FUNCTION - FUNCTIONS - PREDICATES - MIN() - MAX() - xexpr - DELETED - NOT DELETED - !DELETED

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Indexing, Tags, Relations, and Views

Pippets used: - PIP-001 Target Selection - MDO-143 Target Selection - MDO-144 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches known canaries. - Do not send this directly to promotion review. - Run a canary-aware evidence review before PIP-003 is allowed to produce a reviewed candidate.

Evidence tokens under review: - INDEX - REINDEX - REBUILD - SET ORDER - SET INDEX - ASCEND - DESCEND - SEEK - FIND - CNX - CDX - LMDB - BUILDLMDB - REL - RELATIONS - ERSATZ - VIEW

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Generated command draft pages remain draft evidence, not final command reference prose. - The section preserves the distinction between logical/user-facing abstractions and physical backend details. - CDX/CNX language should be reviewed against current implementation and HELP/META evidence before final wording. - SET-family canonicalization remains deferred unless separately repaired or accepted. - Relations, tuple traversal, browser rendering, and views must not be collapsed into one ownership model.

Purpose of this section

This section explains the concepts that control ordered traversal, indexed lookup, relation-aware traversal, and view/projection terminology in DotTalk++.

It follows Navigation, Browsing, and Search because that section deliberately deferred index-specific and relation-specific behavior. Navigation can say that traversal and search may depend on context. This section explains the important context: active order, tags, logical index surfaces, physical backend boundaries, relations, and views.

The goal is not to publish a final index command reference. The goal is to establish safe developer-manual prose that keeps ownership boundaries visible and prevents logical abstractions from being confused with physical backends.

Evidence lanes

This draft uses several evidence lanes.

Current DotTalk evidence lane: - INDEX - REINDEX - REBUILD - SET ORDER - SET INDEX - ASCEND - DESCEND - SEEK - FIND - CNX - CDX - LMDB - BUILDLMDB - REL - RELATIONS - ERSATZ - VIEW

Generated command-reference lane: - Generated command pages may identify available draft command evidence. - They are not final prose and should not be quoted as final manual authority. - Duplicate commands, aliases, slug collisions, and SET-family canonicalization still require command-reference review.

Concept lane: - Indexing provides ordered or keyed access paths. - Tags name or select logical orders. - Active order affects traversal and key-style search where supported. - Relations connect areas or tables through traversal rules. - Views and browser output are projection surfaces unless proven otherwise.

Compatibility lane: - xBase/FoxPro lineage can explain vocabulary, but compatibility material must not be promoted as current DotTalk behavior without runtime proof. - SET-family commands are especially compatibility-sensitive.

Future META feeder lane: - SYSCMD should eventually carry command identity and handler alignment. - SYSSUB-CMD should eventually carry SET ORDER, SET INDEX, REL, and related subcommand identity. - SYSENTVAR should eventually carry aliases, variants, and shortcut spellings after seed hygiene review. - SYSARGS should eventually carry tag names, key expressions, relation/view arguments, and rebuild options. - SYSMSG should eventually carry missing tag, order not found, backend build, relation warning, and rebuild diagnostics. - SYSHELP should eventually carry curated/generated concept help for indexes, tags, relations, views, and traversal.

Indexing vocabulary

Indexing vocabulary needs careful separation.

A user may think about indexes as a way to find records faster or view records in a useful order. A developer manual should be more precise: an index or order is a traversal/access structure that can influence how commands move through or locate records.

Important terms: - index: a structure or command surface associated with ordered/keyed access. - order: the currently selected traversal order where supported. - tag: a named logical order within a multi-tag abstraction. - rebuild/reindex: operations that refresh or rebuild index structures. - key expression: the value or expression used to derive ordered/keyed lookup behavior.

The section should avoid claiming that every navigation or search command always uses an index. That behavior must be proven per command and context.

Open index architecture note

The indexing lane is intentionally open enough to teach and inspect.

That means the manual should preserve the distinction between:

- canonical runtime indexing surfaces such as CNX, CDX, active order, and rebuild paths
- physical backend boundaries such as LMDB for x64 storage
- educational or lab-facing index experiments such as INX, SCX, or SIX

The important manual rule is not to flatten those families into one generic word. They overlap conceptually, but they do not all serve the same role.

Logical order and active order

The active order is a current traversal context. It may affect display, movement, and search behavior depending on the command path.

A safe explanation is: - table state has a current record position; - ordered traversal may change the sequence in which records are visited; - commands such as LIST can expose the actual traversal order; - key-style commands such as SEEK or FIND may depend on active order or key context where supported.

Known canary: - Reported active order must agree with actual traversal order before an order path is marked proven. - A command saying an order is active is not enough. LIST, SEEK, rebuild, and runtime smoke evidence must agree.

Ownership reminder

Index systems own:

- tag metadata
- active-order semantics
- rebuild and verification semantics
- attach/open behavior

Projection layers such as LIST, BROWSE, ERSATZ, GUI, and TUI may reveal ordered traversal, but they do not own index truth. They are evidence surfaces, not index authorities.

Tags and tag availability

A tag names a logical order. In a multi-tag model, selecting a tag should select a user-facing logical order.

This section should preserve this rule: - tag availability and reported active tag must be verified against actual traversal behavior.

Known proof-sensitive case: - A tag may be reported as selected before it is actually available or rebuilt. - After rebuild, traversal may become consistent. - This should remain a canary until runtime proof closes it.

Do not hide this from developer documentation. The manual may keep user-facing prose simple, but the Developer Manual must preserve the proof boundary.

CDX, CNX, and LMDB boundary

The preferred doctrine is:

- CDX/CNX are logical or user-facing index abstractions.
- LMDB is a physical backend and should remain hidden from ordinary command-surface prose unless the section is explicitly backend/developer-facing.

This does not mean LMDB is unimportant. It means ordinary command documentation should avoid making users think they are operating directly on the physical backend when they are really selecting orders, tags, or logical index structures.

Developer-facing prose may explain: - the logical abstraction seen by commands; - the physical backend used internally; - where the backend boundary must not leak into public command vocabulary.

This boundary should be reviewed against current source and HELP/META evidence before final manual promotion.

SET-family boundary and canonicalization canary

SET-family commands are known to require careful canonicalization.

The section may mention: - SET ORDER - SET INDEX - SET-family command surfaces - the need to distinguish aliases, variants, and canonical commands

But it should not resolve canonicalization casually.

Known canary: - SET-family canonicalization remains deferred. - Generated command-reference pages for SET-family items should remain draft evidence until canonicalization is repaired or explicitly accepted.

Manual rule: - Keep SET-family wording conservative. - Do not treat duplicate or variant generated command pages as final canonical command identity.

SEEK and FIND active-order boundary

Navigation, Browsing, and Search introduced SEEK and FIND as search vocabulary but deferred index-specific behavior. This section owns the active-order boundary for those commands.

A safe statement is: - SEEK and FIND are key-style search commands whose exact behavior may depend on active order, tag, or index context. - The final manual should attach runtime proof before claiming exact behavior. - If a command falls back to physical order, reports active order incorrectly, or requires rebuild first, that must remain visible as a canary.

This section should not collapse SEEK/FIND with LOCATE. LOCATE is more naturally predicate-oriented and belongs with expression/predicate search behavior, even if it appears in navigation prose.

Reindexing and rebuild behavior

REINDEX, REBUILD, and BUILDLMDB touch refresh/build behavior.

A conservative explanation is: - rebuild/reindex commands update index or backend structures; - their exact scope and backend effects must be verified by command evidence; - public-facing wording should focus on refreshing logical orders or tags; - backend-specific wording belongs in developer/backend notes.

Known risk: - BUILDLMDB and LMDB terminology can pull physical storage details into user-facing prose. - Keep public/user wording logical unless a developer/backend section explicitly opens the physical layer.

Relations and relation traversal

Relations connect work areas or tables so that a parent context can lead to related child records.

This section should explain relation traversal without making browser output the owner of relation semantics.

Ownership rule: - relation subsystem owns relation definitions and traversal intent; - tuple infrastructure owns relation-aware row projection; - browser and ERSATZ surfaces render or navigate projected relation state; - workspace/session systems own restored area and relation context.

This section can cross-reference the Workspaces and Tuple sections rather than fully restating them.

Known proof: - MCC/x32 relation paths and ERSATZ browser output have provided useful runtime evidence. - x64 workspace/ERSATZ load reporting remains canary-sensitive.

Views and projection boundary

The word view is dangerous because it can mean a saved query, a projection, a browser surface, or a conceptual display. The manual should not assume one meaning until HELP/META/source evidence classifies the actual command surface.

Safe language: - a view/projection presents selected or arranged data; - projection is not storage ownership; - relation-aware projection is not the same as relation definition; - browser output is not the same as table or relation ownership.

This section should preserve that ambiguity until the command-reference and source evidence clarify VIEW and related surfaces.

ERSATZ and browser caution

ERSATZ and relation-aware browser output are valuable evidence surfaces, but they are path-specific.

Known cautions: - plain ERSATZ/no-arg paths and MCC/x32 relation browser paths have useful proof value; - ERSATZ GRID was previously deferred because its snapshot branch did not preserve the same complete Browser-Snapshot; - browser output may be usable even when workspace load reporting is noisy; - projection output should not be promoted to semantic ownership.

Use ERSATZ evidence to explain what can be seen. Do not use it alone to prove every underlying workspace/relation/load behavior.

Known canaries

This section must keep the following canaries visible:

- SET-family canonicalization is deferred.
- SET ORDER and active tag reporting must agree with actual traversal before order behavior is marked proven.
- CDX/CNX must remain logical/user-facing abstractions unless developer/backend context is explicit.
- LMDB is a physical backend and should not leak into ordinary command-surface prose.
- SEEK/FIND active-order behavior requires runtime proof.
- Generated command pages remain draft evidence.
- Relations, tuple traversal, browser rendering, and views require separate ownership language.
- ERSATZ/browser evidence is path-specific.
- x64 workspace/ERSATZ load reporting remains canary-sensitive.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- SET-family canonicalization
- SET ORDER active tag reporting
- CDX/CNX logical abstraction
- LMDB physical backend boundary
- SEEK/FIND active order dependency
- relation tuple browser ownership
- ERSATZ path-specific evidence
- x64 workspace ERSATZ load reporting

These anchors preserve the canaries that the prose already discusses in ordinary language. They should remain until the section is promoted through evidence review. ### Future META alignment

This section should eventually align with the metadata system.

Expected future feeders: - SYSCMD for command identity and handler mapping. - SYSSUBCMD for SET ORDER, SET INDEX, REL, and other subcommand identity. - SYSENTVAR for aliases, variants, and shortcuts after seed hygiene review. - SYSARGS for tag names, key expressions, relation names, view arguments, rebuild flags, and backend options. - SYSMSG for diagnostics about missing tags, unavailable orders, not-found search results, relation warnings, backend build failures, and rebuild results. - SYSHELP for curated/generated concept help about indexing, tags, relations, views, and traversal.

Temporary evidence is acceptable only when marked as temporary and crosswalked to future META feeders.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- no placeholder markers remain;
- all required tokens are represented or intentionally excluded;
- generated command pages are treated as draft evidence;
- SET-family canonicalization is not resolved by prose alone;
- CDX/CNX/LMDB boundary is preserved;

- SEEK/FIND behavior is not overclaimed;
- relation, tuple, browser, and view ownership are not collapsed;
- canaries are present and explicit;
- future META feeders are named;
- no compatibility evidence is presented as runtime proof;
- no backend implementation detail is accidentally promoted into user command prose.

Recommended required tokens for later PIP-003: - INDEX - REINDEX - REBUILD - SET ORDER - SET INDEX - ASCEND - DESCEND - SEEK - FIND - CNX - CDX - LMDB - BUILDLMDB - REL - RELATIONS - ERSATZ - VIEW

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Messages, Errors, and Diagnostics

Pippets used: - PIP-001 Target Selection - MDO-159 Target Selection - MDO-160 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches message-catalog, HELP, metadata, validation, and runtime-reporting boundaries. - Do not send this directly to generic PIP-003. - Run a slow-lane message/diagnostic review first.

Evidence tokens under review: - SYSMMSG - SYSTEM_MESSAGES - MSG_ID - SYMBOL - ENUM_NAME - SEVERITY - FACILITY - SHORT_TXT - SUG_ACT - HELP - HELP GIANT - CMDHELPCHK - WARNING - ERROR - STATUS - SHARED_MSG - diagnostic - message catalog - typed message - parser warning - nonnumeric aggregate - no active table - not found

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Generated command draft pages remain draft evidence, not final command prose. - SYSMMSG is preserved as a future typed message feeder even if current metadata seeding is sparse. - SYSTEM_MESSAGES may represent an older or alternate long-form metadata schema and must be crosswalked carefully. - HELP explains and CMDHELPCHK validates; neither should be treated as runtime proof. - Message catalog doctrine must not mutate HELP, META, CMDHELPCHK, catalogs, source, or runtime data during manual draft assembly.

Purpose of this section

This section explains how DotTalk++ developer documentation should treat messages, errors, warnings, statuses, diagnostics, and future typed message catalog work.

It follows the expression and aggregate section because that section introduced several diagnostic-heavy cases: nonnumeric aggregate values, parser ambiguity, empty or deleted-only aggregate inputs, no-active-table conditions, not-found outcomes, and expression/function argument errors. Those cases need a shared diagnostic vocabulary before the manual expands deeper into command families and subsystem behavior.

The goal is not to claim that the full typed messaging system is already complete. The goal is to preserve the intended direction while keeping current proof boundaries visible.

Evidence lanes

This draft uses several evidence lanes.

Current HELP evidence lane: - HELP - HELP GIANT - HELP_LINE - HELP_ARTIFACTS - WARNING rows - ERROR rows - STATUS rows - SHARED_MSG rows

Current metadata evidence lane: - SYSMMSG - SYSTEM_MESSAGES - SYSHELP - SYSCMD - SYSSUBCMD - SYSFUNC - SYSARGS - SYSENTVAR

Diagnostic concept lane: - message - diagnostic - warning - error - status - trace - log - test output - parser warning - message catalog - typed message

Runtime/source proof lane: - parser warnings require runtime/source evidence; - expression errors require runtime/source evidence; - aggregate errors require runtime/source evidence; - no-active-table messages require runtime/source evidence; - not-found messages require runtime/source evidence.

Future catalog lane: - the messaging layer should become the single typed, catalog-backed reporting path; - current coverage must be verified before final manual claims; - sparse metadata seeding is a future feeder, not a reason to ignore the lane.

Authority boundaries

The same doctrine applies here:

- Runtime proves behavior.
- Source defines implementation and subsystem ownership.
- HELP explains behavior and command surface intent.
- Metadata organizes identity, ownership, argument shapes, help text, and message catalog alignment.
- CMDHELPCHK validates HELP/catalog consistency.
- SelfDoc preserves provenance.
- Manualgen assembles.

This section must be careful because HELP and diagnostics are tempting to treat as proof. HELP output can explain intended or cataloged behavior, but it does not by itself prove runtime execution. CMDHELPCHK can validate consistency, but it does not by itself prove runtime behavior either.

Message vocabulary

The manual should not collapse every output line into the same kind of message.

Useful distinctions: - A message is a reported unit of text or structured information. - A diagnostic explains a condition that may require interpretation or action. - A warning reports a risk or questionable condition. - An error reports a failed operation or invalid condition. - A status reports state or completion information. - A trace reports internal execution evidence. - A log records events for later inspection. - A test result reports expected versus observed behavior. - HELP text explains usage, concepts, warnings, examples, and reference material. - A catalog row organizes message identity and metadata.

These categories may overlap in implementation, but the manual should not collapse them until source and metadata evidence prove the ownership model.

Typed message catalog direction

The project direction is that the messaging layer should become the single typed, catalog-backed reporting path for: - commands; - errors; - help text; - syntax issues; - warnings; - traces; - UI/status messages; - logs; - tests; - HELP validation; - upper-layer metadata reporting and collection.

This is directionally important, but not automatically complete. The manual should say: - intended direction: typed, catalog-backed reporting; - current proof: must be verified by source, runtime, HELP, and metadata evidence; - draft boundary: no HELP/META/CMDHELPCHK/catalog/source/runtime mutation during manual assembly.

SYMSMSG and SYSTEM_MESSAGES

SYMSMSG is the compact/current metadata feeder identified for message catalog work. Its fields include message identity and message metadata such as: - MSG_ID - SYMBOL - ENUM_NAME - SEVERITY - FACILITY - SHORT_TXT - IMPL_STAT - VIS_TIER - OWNER - SRC_AUTH - SRC_FILE - PUB_SURF - USED_RUN - ACTIVE - VER_AT - SUG_ACT - NOTES

SYSTEM_MESSAGES appears in earlier or alternate metadata material as a long-form schema name. It should not be assumed identical to SYMSMSG without a crosswalk.

Safe wording: - SYMSMSG is the current compact future feeder for message metadata where seeded and verified. - SYSTEM_MESSAGES may be legacy or alternate long-form metadata evidence. - The manual should crosswalk them carefully before treating them as one authority.

HELP rows as diagnostic evidence

HELP GIANT and HELP tables provide useful evidence for diagnostic text and categories.

Evidence examples: - WARNING rows; - ERROR rows; - STATUS rows; - SHARED_MSG rows; - SOURCE_FACT rows; - HELP_LINE records; - HELP_ARTIFACTS records.

But HELP rows are explanation/catalog evidence, not runtime proof. A WARNING row in HELP means the help system has a warning artifact; it does not prove the runtime command currently emits that warning.

SHARED_MSG caution

SHARED_MSG rows are useful because they suggest text or messages shared across HELP or command surfaces.

Safe wording: - SHARED_MSG rows are evidence. - They may be sparse. - They should not be treated as complete message catalog coverage. - They should be crosswalked to SYMSMSG/SYSTEM_MESSAGES and source/runtime evidence before final implementation claims.

CMDHELPCHK role

CMDHELPCHK validates HELP/catalog consistency. It is a validator and system contract checker, not just documentation.

Safe wording: - CMDHELPCHK can identify gaps, inconsistencies, or contract drift. - CMDHELPCHK supports manual assembly by checking HELP/catalog alignment. - CMDHELPCHK does not replace runtime proof. - CMDHELPCHK does not replace source ownership.

This matters because manual assembly can use HELP/META/CMDHELPCHK-first workflow, but truth authority remains runtime/source/HELP/metadata/CMDHELPCHK according to their roles.

Runtime diagnostic examples

Expressions and aggregates introduced diagnostic examples that should eventually map into message evidence: - nonnumeric aggregate expression; - character expression used with AVG; - empty or deleted-only aggregate input; - invalid field or expression; - parser ambiguity between command and function form; - no active table; - not found; - unsupported command syntax; - missing argument; - invalid argument count.

This draft does not claim all of those are cataloged today. It says they are natural future candidates for SYMSMSG and source/runtime review.

Severity vocabulary

Severity vocabulary must not be invented.

Possible severity words include: - ERROR - WARNING - STATUS - INFO - TRACE - DEBUG

But final wording should only claim severity categories when evidence supports them through SYMSMSG, HELP, source, or runtime behavior. The draft should avoid inventing a complete severity taxonomy.

Parser warnings and syntax diagnostics

Parser warnings and syntax diagnostics are important because command-line behavior can bridge commands and functions. Examples include: - command/function ambiguity; - unknown command; - missing argument; - unsupported syntax; - scalar function form versus aggregate command form.

The expressions section preserved MIN/MAX parser ambiguity. This diagnostics section should explain that such cases need typed messages or at least consistent diagnostic reporting, but it should not claim the final routing is complete without source/runtime proof.

No-active-table and not-found messages

No-active-table and not-found messages are common diagnostic candidates.

They should be handled conservatively: - no-active-table behavior depends on command context; - not-found behavior depends on search command, index/order context, and runtime state; - exact wording and severity need runtime/source evidence; - future SYMSG rows should eventually organize the message identities.

Message catalog and HELP alignment

Message catalog work should eventually align: - SYMSG message identity; - HELP text; - command/function ownership; - argument validation; - runtime emission; - CMDHELPCHK validation; - SelfDoc provenance.

This section should preserve that alignment goal but avoid claiming it is finished.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- typed catalog-backed reporting path coverage
- HELP CMDHELPCHK not runtime proof
- SYMSG SYSTEM_MESSAGES schema variation
- SHARED_MSG sparse evidence
- parser expression aggregate no-active-table not-found runtime evidence
- diagnostics warnings errors statuses traces logs tests help catalog distinction
- SYMSG future feeder sparse seed
- no mutation during manual message catalog draft assembly
- severity vocabulary not invented
- typed-message ownership future catalog alignment

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Future META alignment

This section should eventually align with the metadata system.

Expected future feeders: - SYMSG for message identity, symbol, enum name, severity, facility, short text, implementation status, visibility tier, owner, source authority, source file, public surface flag, used-at-runtime flag, suggested action, notes, and active status. - SYSTEM_MESSAGES for legacy or alternate long-form message schema evidence that must be crosswalked carefully. - SYSHELP for help text connected to message owners, diagnostic concepts, generated text, and curated text. - SYSCMD for command owners that emit diagnostics. - SYSSUBCMD for subcommand diagnostic surfaces when command families own messages. - SYSFUNC for function-related diagnostics such as argument count, nonnumeric value, parser ambiguity, and calculation errors. - SYSARGS for arguments involved in diagnostic validation and error reporting. - SYSENTVAR for aliases or variants that may affect diagnostic routing. - HELP_LINE and HELP_ARTIFACTS as current HELP evidence lanes for WARNING, ERROR, STATUS, SHARED_MSG, and related text.

Temporary evidence is acceptable only when marked as temporary and crosswalked to future META feeders.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- generated command pages are treated as draft evidence;
- HELP explains and CMDHELPCHK validates, but neither is presented as runtime proof;
- SYMSG and SYSTEM_MESSAGES are not collapsed casually;
- SHARED_MSG is treated as sparse evidence;
- diagnostic categories are separated;
- severity vocabulary is not invented;
- parser warnings and expression/aggregate diagnostics are runtime/source gated;
- SYMSG remains a future feeder even if sparse;
- no manual draft work mutates HELP, META, CMDHELPCHK, catalogs, source, or runtime data.

Recommended required tokens for later PIP-003: - SYMSG - SYSTEM_MESSAGES - MSG_ID - SYMBOL - ENUM_NAME - SEVERITY - FACILITY - SHORT_TXT - SUG_ACT - HELP - HELP GIANT - CMDHELPCHK - WARNING - ERROR - STATUS - SHARED_MSG - diagnostic - message catalog - typed message - parser warning - nonnumeric aggregate - no active table - not found

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

HELP, Metadata, CMDHELPCHK, and Manualgen Alignment

Pippets used: - PIP-001 Target Selection - MDO-173 Target Selection - MDO-174 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches truth authority doctrine, manual assembly workflow, HELP evidence, metadata evidence, CMDHELPCHK validation, SelfDoc provenance, source authority, and future feeder alignment. - Do not send this directly to generic PIP-003. - Run a slow-lane authority/alignment review first.

Evidence tokens under review: - HELP - HELP GIANT - HELP_LINE - HELP_ARTIFACTS - CMDHELP - CMDHELPCHK - META - metadata - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - SYMSG - SYSFUNC - SOURCE - SOURCE_FACT - SelfDoc - manualgen - PIP - validator - crosswalk - report-only - truth authority - assembly workflow

Draft notes: - This is conservative manual prose for slow-lane evidence review. - HELP/META/CMDHELPCHK-first is a manual assembly workflow, not a replacement for truth authority. - Source remains implementation authority even when manualgen starts from HELP, metadata, and CMDHELPCHK. - Sparse metadata feeders are future alignment lanes, not dead ends. - Temporary evidence lanes must be labeled and crosswalked to future META feeders. - Manualgen and SelfDoc work remain report-only unless explicitly authorized to mutate production artifacts.

Purpose of this section

This section explains how the Developer Manual should align HELP, metadata, CMDHELPCHK, SelfDoc, and manualgen without confusing their roles.

It follows the command-surface section because recent sections repeatedly relied on the same evidence pattern. Manualgen has been reading HELP broadly, reading metadata semantically, validating with CMDHELPCHK, verifying with source, proving with runtime, and assembling manual sections. That is a useful assembly workflow. It is not a replacement for truth authority.

The goal of this section is to prevent drift. Readers should understand which evidence lane explains, which organizes, which validates, which proves, which defines implementation, and which assembles manual output.

Core doctrine

The manual should use this doctrine consistently:

- Runtime proves behavior.
- Source defines implementation and subsystem ownership.
- HELP explains behavior, vocabulary, command usage, examples, and concepts.
- Metadata organizes identity, ownership, arguments, variants, messages, and alignment.
- CMDHELPCHK validates HELP/catalog consistency and detects drift.
- SelfDoc preserves provenance and report-only evidence.
- Manualgen assembles drafts, reviews, gates, and promoted draft workspaces.

The doctrine matters because each lane is useful but limited. HELP can be excellent explanatory evidence without proving runtime behavior. Metadata can organize a future target even when currently sparse. CMDHELPCHK can catch alignment drift without replacing runtime tests or source review.

Assembly workflow versus truth authority

Manual assembly can use a HELP/META/CMDHELPCHK-first workflow:

- read HELP broadly;
- read META semantically;
- validate with CMDHELPCHK;
- verify with source;
- prove with runtime;
- assemble with manuals.

That workflow is practical because HELP, metadata, and CMDHELPCHK give manualgen a wide map of the system. But it must not be described as the truth authority doctrine.

Truth authority remains role-separated:

- runtime proves observed behavior;
- source defines implementation and subsystem ownership;
- HELP explains;
- metadata organizes;
- CMDHELPCHK validates.

Safe wording: - HELP/META/CMDHELPCHK-first is assembly order. - It is not authority order. - Source is not demoted to a sidecar outside implementation truth. - Runtime proof remains required for behavior claims.

HELP lane

HELP is one of the strongest manual assembly feeders.

Current HELP evidence may include: - HELP topics; - HELP GIANT output; - HELP_LINE rows; - HELP_ARTIFACTS rows; - CMDHELP artifacts; - generated HELP reports; - usage text; - warnings; - examples; - concept notes.

HELP explains the command surface and conceptual model. It can also expose vocabulary that manualgen should review.

Safe wording: - HELP explains. - HELP may reveal intended behavior or documented behavior. - HELP should not be treated as runtime proof. - HELP should be crosswalked to source, runtime, metadata, and CMDHELPCHK where claims matter.

Metadata lane

Metadata organizes the system.

Current and future metadata feeders include: - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - SYSMMSG - SYSFUNC - SOURCE_FACT - HELP_LINE - HELP_ARTIFACTS

Metadata may be sparse, partially seeded, or split between older and newer schema forms. Sparse metadata is not a reason to ignore the lane. Sparse metadata should be labeled as a future alignment feeder until seeded and verified.

Safe wording: - Metadata organizes identity and relationships. - Metadata does not automatically prove runtime behavior. - Sparse metadata tables are future feeders, not dead ends. - Alternate metadata schemas should be crosswalked rather than collapsed casually.

CMDHELPCHK lane

CMDHELPCHK is a validator and system contract checker.

It can: - identify missing or inconsistent HELP rows; - compare command surfaces against HELP/catalog evidence; - report drift; - support manual assembly gates; - validate generated HELP and metadata alignment.

Safe wording: - CMDHELPCHK validates. - CMDHELPCHK can identify drift and gaps. - CMDHELPCHK does not prove runtime execution. - CMDHELPCHK does not replace source ownership.

SelfDoc lane

SelfDoc preserves provenance.

SelfDoc evidence includes: - source-comment contracts; - harvested source evidence; - report-only metadata staging; - provenance reports; - canary ledgers; - source/miner evidence; - manualgen run records; - savepoint journals.

SelfDoc should keep evidence visible and traceable. It should not silently mutate production HELP, metadata, CMDHELPCHK, catalogs, source, or runtime data during draft assembly.

Safe wording: - SelfDoc preserves provenance. - SelfDoc defaults to report-only. - SelfDoc can identify evidence and drift. - SelfDoc changes to production artifacts require explicit authorization.

Manualgen lane

Manualgen assembles.

Manualgen creates: - draft prose; - reviewed candidates; - promoted draft workspaces; - pipet run records; - summary reports; - gate reports; - savepoint records; - package bundles.

Manualgen does not publish final manuals by itself. Promoted draft workspaces are still draft workspaces unless final publication is explicitly authorized.

Safe wording: - Manualgen assembles. - Manualgen gates and records decisions. - Manualgen does not mutate HELP, META, CMDHELPCHK, catalogs, source, or runtime data during manual draft assembly. - Manualgen promoted drafts are not final publication.

Source lane

Source defines implementation and subsystem ownership.

Source evidence is required when the manual makes claims about: - implementation ownership; - parser routing; - command handlers; - backend ownership; - relation traversal semantics; - expression evaluation ownership; - memo payload lifecycle; - message emission; - runtime behavior that is not proven by a direct test.

Source may be consulted later in the workflow, but later does not mean weaker. In manual assembly order, source may be a verification lane. In truth authority, source remains implementation authority.

Safe wording: - Source defines implementation. - Source verifies ownership and routing. - Source is not merely a provenance sidecar. - Source claims should be tied to files, comments, contracts, or source-miner evidence.

Runtime lane

Runtime proves behavior.

Runtime evidence includes: - observed command runs; - smoke tests; - shakedown transcripts; - exact command output; - pass/fail test logs; - canary reproduction; - before/after behavior.

Runtime evidence is especially important for: - command execution; - parser ambiguity; - error output; - no-active-table behavior; - not-found behavior; - deleted-record behavior; - relation traversal; - memo backend attachment; - index/order behavior.

Safe wording: - Runtime proves observed behavior. - Runtime proof should include concrete commands and outputs where possible. - Runtime proof should be dated or tied to a build/session when possible. - Runtime proof does not by itself describe implementation ownership without source.

Temporary evidence lanes

Manualgen may need to use temporary evidence when future metadata feeders are sparse.

Temporary evidence examples: - generated command pages; - HELP GIANT exports; - current HELP rows; - manually curated canary notes; - runtime shakedown notes; - older metadata schema files; - seed scripts; - user/MDO handoff notes; - source-contract reports; - manualgen pipet reports.

Temporary evidence is allowed when labeled.

Safe wording: - This source is temporary evidence for the current manual pass. - Future feeder should be SYSFUNC, SYSMMSG, SYSCMD, SYSSUBCMD, SYSENTVAR, SYSARGS, SYSHELP, or related metadata. - Temporary evidence should be crosswalked to future META feeders when the metadata system matures. - Temporary evidence must not be promoted as final authority without verification.

Future META alignment

This section should explicitly preserve future feeder alignment.

Expected future feeders: - SYSCMD for command identity, handler alignment, visibility, public surface, display reach, owner, source authority, source file, and help topic. - SYSSUBCMD for subcommand and command-family identity. - SYSENTVAR for aliases, variants, compatibility spellings, shortcut forms, and entry points. - SYSARGS for argument shapes, predicates, filters, scopes, validation surfaces, and repeatable flags. - SYSHELP for curated and generated help text. - SYSMMSG for diagnostics, warnings, statuses, parser messages, and typed message catalog alignment. - SYSFUNC for function identity, categories, argument ranges, handler links, CALC/CALCWRITE reach, public surface, self-registration, and function-command bridge surfaces. - HELP_LINE and HELP_ARTIFACTS for current HELP evidence lanes. - SOURCE_FACT and source-contract evidence for source/comment provenance. - manualgen reports and PIP records for assembly provenance and gate evidence.

Sparse feeders should be kept visible. The manual should not ignore SYSFUNC, SYSMMSG, SYSCMD, SYSSUBCMD, SYSENTVAR, SYSARGS, or SYSHELP because they are empty or only partially seeded today.

Crosswalk discipline

Crosswalks prevent drift.

Useful crosswalks include: - HELP topic to command identity; - command identity to SYSCMD; - subcommand to SYSSUBCMD; - alias or entry variant to SYSENTVAR; - argument shape to SYSARGS; - help text to SYSHELP; - diagnostic text to SYSMMSG; - function reference to SYSFUNC; - source comment to SOURCE_FACT; - generated command page to canonical command and variant review; - manualgen section to evidence tokens and pipet reports.

Crosswalks should preserve uncertainty. A crosswalk can say “candidate match” or “future feeder” without claiming final authority.

Safety boundaries

This alignment section should restate the safety boundaries.

Default manualgen/SelfDoc boundary: - no generated command page deletion; - no HELP mutation; - no META mutation; - no CMDHELPCHK mutation; - no catalog apply; - no source edits; - no production SelfDoc metadata promotion; - no final publication without explicit authorization.

Report-only work is the default.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- help meta cmdhelpchk first is assembly workflow not truth authority
- runtime source help metadata cmdhelpchk truth authority roles
- help explains not runtime proof
- metadata sparse feeders not dead ends
- cmdhelpchk validates not runtime source proof
- sysfunc sysmsg syscmd syssubcmd sysentvar sysargs syshelp sparse feeder alignment
- selfdoc provenance report-only boundaries
- manualgen assembles no mutation
- temporary evidence lanes labeled crosswalked
- source remains implementation authority

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- HELP/META/CMDHELPCHK-first is framed as assembly workflow, not truth authority;
- runtime/source/HELP/metadata/CMDHELPCHK authority roles are preserved;
- HELP is not treated as runtime proof;
- metadata sparse feeders remain visible;
- CMDHELPCHK is not treated as runtime/source proof;
- SYSFUNC, SYSMSG, SYSCMD, SYSSUBCMD, SYSENTVAR, SYSARGS, and SYSHELP are included as future/current feeders;
- SelfDoc report-only provenance boundaries are visible;
- manualgen no-mutation boundary is visible;
- temporary evidence lanes are labeled and crosswalked;
- source remains implementation authority.

Recommended required tokens for later PIP-003: - HELP - HELP GIANT - HELP_LINE - HELP_ARTIFACTS - CMDHELP - CMDHELPCHK - META - metadata - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - SYSMSG - SYSFUNC - SOURCE - SOURCE_FACT - SelfDoc - manualgen - PIP - validator - crosswalk - report-only - truth authority - assembly workflow

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion

- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Command Reference Assembly, Aliases, and Generated Page Hygiene

Pippets used: - PIP-001 Target Selection - MDO-188 Target Selection - MDO-189 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches generated command pages, duplicate command rows, aliases, entry variants, canonical command identity, slug collisions, LOAD guard behavior, SET-family canonicalization, AGGS exposure, and command-reference publication readiness. - Do not send this directly to generic PIP-003. - Run a slow-lane command-reference hygiene review first.

Evidence tokens under review: - command reference - generated command pages - HELP_COMMANDS - HELP - HELP GIANT - CMDHELP - CMDHELPCHK - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - alias - variant - canonical command - slug collision - duplicate command - APPEND BLANK - APPEND_BLANK - LOAD guard - SET-family - AGGS - internal owner - public surface - manualgen - PIP - crosswalk - report-only - no mutation

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Generated command pages are draft evidence, not final command reference prose. - Duplicate command rows and slug collisions must be reviewed before publication. - Aliases and entry variants must not be treated as canonical commands without review. - APPEND BLANK and APPEND_BLANK may map to the same slug but may represent variant or canonicalization evidence. - LOAD guard must preserve no top-level LOAD page while preserving scoped CODASYL LOAD and WORKSPACE LOAD. - SET-family canonicalization remains deferred unless separately repaired or accepted. - AGGS remains internal or family-owner evidence unless explicitly accepted as public command surface. - HELP, CMDHELPCHK, and metadata can guide command reference review but cannot settle runtime or source canaries alone. - Command reference hygiene packages must not delete generated command pages during manual draft assembly.

Purpose of this section

This section explains how the Developer Manual should treat generated command pages, command-reference assembly, aliases, variants, canonical commands, and generated-page hygiene.

It follows the Command Surface, HELP/Metadata/CMDHELPCHK Alignment, and Runtime Evidence sections because those sections established the boundaries needed here. Command reference work needs all three: a model of command surface and entry variants, an alignment model for HELP and metadata, and an evidence model for runtime/source/canary closure.

The goal is not to publish the final command reference in this section. The goal is to define safe rules for assembling the future command reference from generated pages and other evidence without losing alias/variant information, publishing internal scaffolding, or deleting generated evidence prematurely.

Authority model for command reference assembly

The same doctrine applies:

- Runtime proves command behavior.
- Source defines implementation and command ownership.
- HELP explains usage and concepts.
- Metadata organizes command identity, arguments, variants, and help alignment.
- CMDHELPCHK validates HELP/catalog consistency.
- SelfDoc preserves provenance.

- Manualgen assembles drafts, reviews, and promoted workspaces.

Generated command pages are an evidence lane. They are not the authority by themselves.

Safe wording: - Generated pages feed review. - Generated pages do not replace review. - Generated pages may expose duplicates, aliases, scaffolding, or internal owners. - Final command-reference prose needs dedupe, alias/variant review, HELP/metadata alignment, and runtime/source checks where behavior or ownership is claimed.

Generated command pages

Generated command pages are useful because they preserve broad command-surface evidence.

They may show: - command names; - aliases; - variants; - generated slugs; - usage text; - HELP extraction; - metadata alignment candidates; - duplicate rows; - command-family scaffolding; - internal owner entries.

But generated pages can also expose draft artifacts.

Safe wording: - Generated command pages are draft evidence. - They are not final command reference prose. - They should be reviewed, deduped, and crosswalked before publication. - They should not be deleted or rewritten during ordinary manual draft assembly.

Duplicate command rows

Duplicate command rows can mean different things.

A duplicate may be: - a true duplicate that should be collapsed later; - an alias; - an entry variant; - a compatibility spelling; - a scoped subcommand; - a generated slug collision; - an internal owner or command-family scaffold; - a public command that shares a root token with another command.

Safe wording: - Duplicate rows are review input. - Duplicate rows should not be collapsed automatically. - A duplicate row is not proof that one page is wrong. - Dedupe should preserve evidence until canonical command, alias, variant, and scope are reviewed.

Aliases and variants

Aliases and variants are not automatically canonical commands.

An alias may be a shortcut spelling. A variant may be a compatibility form, app-style entry point, scoped command form, or user-facing convenience form. The command reference should preserve those distinctions until metadata and evidence review decide how they should be presented.

Safe wording: - Aliases and variants must not be treated as canonical commands without review. - SYSENTVAR should eventually organize aliases, variants, shortcut spellings, compatibility forms, generated page entries, and app-style entry points. - Generated command pages may represent aliases or variants rather than separate canonical commands. - Manual prose should avoid treating every distinct token as a separate command until dedupe and variant review are complete.

Canonical command identity

Canonical command identity is the reviewed identity used for final reference organization.

A canonical command should eventually align with: - SYSCMD command identity; - handler or source ownership where relevant; - public surface status; - help topic; - argument model; - aliases and variants; - generated page crosswalk; - runtime/source evidence where behavior or ownership is claimed.

Safe wording: - Canonical command identity should be reviewed. - Canonical command identity should not be inferred from slug text alone. - Canonical command identity should not erase useful alias or variant evidence. - Metadata can organize identity, but runtime/source evidence may still be required for behavior and ownership claims.

Slug collisions

A slug collision occurs when two different generated entries map to the same or confusing page slug.

Slug collisions may happen because: - spaces and underscores collapse; - punctuation is normalized; - aliases share words; - variants differ only by formatting; - scoped command names are flattened; - generator rules remove important distinctions.

Safe wording: - Slug collisions are review items. - Slug collisions should not be resolved by deleting generated pages during manual draft assembly. - Slug collision repair should preserve source evidence, generated evidence, and future metadata crosswalks. - A later command-reference hygiene pass may choose canonical slugs after review.

APPEND BLANK and APPEND__BLANK

APPEND BLANK and APPEND__BLANK are useful review examples.

They may map to the same slug or appear related in generated evidence, but the manual should not force a final answer here. They may represent: - a command plus argument form; - a compatibility form; - a generated normalization issue; - an alias or variant; - a canonicalization candidate.

Safe wording: - APPEND BLANK and APPEND__BLANK should remain variant/canonicalization review examples. - Do not treat one as wrong solely because of slug shape. - Do not publish a final command-reference rule until evidence review decides canonical command, variant, and argument treatment.

LOAD guard

LOAD is a known generated-page guard canary.

The guard should preserve: - no top-level LOAD page where that surface is intentionally suppressed; - scoped CODASYL LOAD where supported; - WORKSPACE LOAD where supported; - evidence rows that explain why top-level LOAD is guarded.

Safe wording: - LOAD guard must preserve no top-level LOAD page while preserving scoped CODASYL LOAD and WORKSPACE LOAD. - LOAD guard is not permission to delete all LOAD-related evidence. - Scoped LOAD forms should remain visible for review if they are evidenced. - A guard is a publication boundary, not an evidence deletion rule.

SET-family canonicalization

SET-family canonicalization remains deferred.

SET-family entries may include: - SET ORDER; - SET INDEX; - other SET-scoped command surfaces; - compatibility forms; - generated pages that flatten or split command/subcommand identities.

Safe wording: - SET-family canonicalization remains deferred unless separately repaired or accepted. - SET-family generated pages should be treated as draft evidence. - Indexing owns order/tag semantics; command-reference hygiene owns command identity and reference organization. - Final reference organization should not be settled by prose alone.

AGGS boundary

AGGS is a known command-reference canary.

Current doctrine: - AGGS is intended as an internal owner or aggregate-family grouping. - Direct aggregate verbs such as SUM, AVG, MIN, and MAX are the intended user-facing aggregate command surface where evidenced. - AGGS appearing in generated command pages may indicate internal owner exposure, scaffold leakage, or family grouping evidence. - AGGS should not be published as a public command surface unless explicitly accepted.

Safe wording: - AGGS remains internal or family-owner evidence unless explicitly accepted as public command surface. - Generated AGGS evidence should be retained for review. - Generated AGGS evidence should not become final public command-reference prose by default.

Internal owner and public surface

Command-reference assembly must separate internal owner evidence from public command surface.

An internal owner may be: - a command-family grouping; - handler scaffolding; - parser dispatch label; - generated metadata row; - development/debug surface; - owner used for help organization.

A public surface is what users are intended to invoke.

Safe wording: - Internal owner evidence is useful. - Internal owner evidence is not automatically public command surface. - Public surface requires intent and evidence. - PUB_SURF, VIS, DISP_REACH, HELP, runtime behavior, and source ownership may all participate in review.

HELP and CMDHELPCHK

HELP and CMDHELPCHK are important command-reference inputs.

HELP can explain: - usage; - concepts; - examples; - warnings; - aliases; - command families.

CMDHELPCHK can validate: - HELP/catalog consistency; - missing topics; - generated HELP drift; - command catalog alignment.

Safe wording: - HELP explains command reference intent. - CMDHELPCHK validates command-reference alignment. - Neither HELP nor CMDHELPCHK replaces runtime proof or source ownership. - HELP and CMDHELPCHK can guide command-reference hygiene.

Metadata feeders

Future and current metadata feeders should remain visible.

Expected feeders: - SYSCMD for canonical command identity, canonical name, handler, visibility, public surface, display reach, owner, source authority, and help topic. - SYSSUBCMD for scoped subcommand identity such as SET ORDER, SET INDEX, scoped LOAD forms, and command-family subcommands. - SYSENTVAR for aliases, variants, shortcut spellings, compatibility forms, generated page entries, and app-style entry points. - SYSARGS for command argument shapes, scopes, deleted filters, predicates, tag names, workspace files, and rebuild options. - SYSHELP for curated and generated command help text. - SYSMSG for parser, syntax, unknown command, invalid argument, and ambiguity diagnostics. - SYSFUNC for function-command bridge cases where scalar functions overlap command-like syntax.

Sparse feeders are still alignment lanes. They should not be ignored because they are incomplete.

Command-reference crosswalks

Crosswalks should connect generated pages to reviewed command identity without collapsing evidence.

Useful crosswalks include: - generated page to SYSCMD canonical command; - generated page to SYSENTVAR alias or variant; - generated page to SYSSUBCMD scoped subcommand; - generated usage to SYSARGS argument model; - HELP topic to SYSHELP; - command diagnostic to SYSMSG; - function-command bridge entry to SYSFUNC; - generated slug to reviewed final slug; - command-family owner to public command entries.

Safe wording: - Crosswalks may be candidate, partial, or verified. - Crosswalks should preserve uncertainty. - Crosswalks should not delete evidence. - Crosswalks help future metadata absorb temporary generated-page evidence.

Publication readiness

A generated command page is not publication-ready by default.

Before a command page becomes final reference prose, review should check: - canonical command identity; - alias and variant treatment; - duplicate row handling; - slug selection; - HELP alignment; - argument model; - public surface status; - internal owner exposure; - runtime behavior where behavior is claimed; - source ownership where ownership is claimed; - metadata crosswalk; - CMDHELPCHK consistency; - no unresolved canaries.

Safe wording: - Command-reference publication is a later gate. - Draft generated pages are review material. - This section does not publish the final reference. - Publication should be guarded by evidence, crosswalks, and human acceptance.

No-delete and no-mutation safety

Command-reference hygiene must preserve generated evidence during manual draft assembly.

Default boundary: - no generated command page deletion; - no HELP mutation; - no META mutation; - no CMDHELPCHK mutation; - no catalog apply; - no source edits; - no runtime data mutation; - no production SelfDoc metadata promotion; - no final publication.

Safe wording: - Review can flag generated pages. - Review can recommend later repair. - Review must not delete generated pages during manual draft assembly. - Production mutation requires explicit authorization.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- generated pages draft evidence not final reference
- duplicate rows slug collisions reviewed before publication
- aliases variants not canonical without review
- append blank append_blank variant canonicalization example
- load guard preserve no top-level load scoped codasyl load workspace load
- set-family canonicalization deferred
- aggs internal owner unless accepted public surface
- help cmdhelpchk metadata guide not runtime source closure
- command reference hygiene no generated page deletion
- sysentvar syscmd syssubcmd sysargs syshelp feeder alignment

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- generated pages are framed as draft evidence, not final reference;
- duplicate rows and slug collisions are review items, not automatic deletion targets;
- aliases and variants are not collapsed into canonical commands;
- APPEND BLANK and APPEND_BLANK remain review examples;
- LOAD guard preserves no top-level LOAD page while preserving scoped forms;
- SET-family canonicalization remains deferred;
- AGGS remains internal/family-owner evidence unless accepted;
- HELP/CMDHELPCHK/metadata are review guides but do not close runtime/source canaries alone;
- no generated command page deletion or production mutation is authorized.

Recommended required tokens for later PIP-003: - command reference - generated command pages - HELP_COMMANDS - HELP - HELP GIANT - CMDHELP - CMDHELPCHK - SYSCMD - SYSSUBCMD - SYSENTVAR - SYSARGS - SYSHELP - alias - variant - canonical command - slug collision - duplicate command - APPEND BLANK - APPEND_BLANK - LOAD guard - SET-family - AGGS - internal owner - public surface - manualgen - PIP - crosswalk - report-only - no mutation

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated

- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Promoted Draft Review, Header Normalization, and Publication Readiness

Pippets used: - PIP-001 Target Selection - MDO-196 Target Selection - MDO-197 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches promoted draft review, review packet practice, candidate note cleanup, reviewed candidate status cleanup, header normalization, path repair, canonical path verification, table of contents checks, section order checks, final publication boundaries, generated command page no deletion, and no mutation safety. - Do not send this directly to generic PIP-003. - Run a slow-lane promoted-draft readiness review first.

Evidence tokens under review: - promoted draft - review packet - inspection packet - human review - candidate note - reviewed candidate status - header normalization - publication readiness - final publication - table of contents - section order - section count - path repair - canonical path - slug - generated command pages - no deletion - HELP - CMDHELPCHK - metadata - SelfDoc - manualgen - PIP - report-only - no mutation

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Promoted draft workspace assembly is not final publication. - Candidate note headers in promoted draft sections must not be normalized casually one section at a time. - Reviewed candidate status blocks in promoted draft sections must be handled by a systematic promoted-draft header normalization pass. - Header normalization must not rewrite substantive prose without explicit review. - Path and slug verification must use canonical section ids and inspectable files. - Table of contents and section order must be verified against actual section files. - Review packets should remain the preferred human-inspection surface before authorization. - Generated command pages and evidence artifacts must not be deleted during publication-readiness work. - Publication readiness may recommend repairs, but it should not silently publish or normalize.

Purpose of this section

This section explains how the Developer Manual should review a promoted draft workspace, prepare for header normalization, and separate publication readiness from final publication.

It follows the Command Reference Assembly section because that section exposed a concrete promoted-section path problem. MDO-194 reported a successful promotion, but MDO-194A was needed to repair and verify the canonical promoted section path. That experience proves that status reports are not enough. A promoted draft must be inspectable, path-checked, section-counted, and reviewed as a workspace.

The goal is not to publish the manual. The goal is to define a safe review lane between promoted draft assembly and any later final publication step.

Promoted draft workspace

A promoted draft workspace is an assembled manual draft.

It may contain: - reviewed candidate prose copied into section files; - candidate note headers; - reviewed candidate status blocks; - promoted section paths; - generated section ordering; - table of contents material or future table of contents inputs; - evidence of promotion history; - known draft-workspace debt.

Safe wording: - A promoted draft is not final publication. - A promoted draft is a reviewable workspace. - A promoted draft can contain review artifacts that are useful during assembly but inappropriate for final publication. - Final publication should require a later explicit gate.

Inspectable files

The user should be able to inspect the actual prose.

Inspectable files matter because status reports can be green while path or filename issues remain. MDO-194A showed why the canonical path must be checked directly.

Safe wording: - Review must include actual section files, not only status CSVs. - Canonical paths should be opened or tested directly. - A readable section file is stronger evidence than a report that only says a section exists. - Review packets should make prose easy to inspect before authorization.

Review packets and inspection packets

Review packets are human-inspection surfaces.

They should provide: - the prose to read; - a checklist; - known canaries; - a summary; - a hold, repair, or accept decision point.

Safe wording: - Review packets should remain the preferred human-inspection surface before authorization. - Review packets should not create human decisions by themselves. - Review packets should not promote. - Review packets should help the reviewer decide HOLD, REPAIR, or ACCEPT_FOR_PROMOTION.

Candidate note headers

Candidate note headers are useful during review.

They preserve: - origin of the candidate; - promotion-gate provenance; - review warning; - draft status.

But candidate note headers can become clutter in a final manual.

Safe wording: - Candidate note headers in promoted draft sections must not be normalized casually one section at a time. - Candidate note headers should be handled by a systematic promoted-draft header normalization pass. - Candidate note headers should not be removed until the manual has a defined normalization rule. - Header normalization should preserve provenance in reports even if public-facing prose is cleaned later.

Reviewed candidate status blocks

Reviewed candidate status blocks are also useful during review.

They say that a section passed a reviewed-candidate lane, but they are not necessarily final publication text.

Safe wording: - Reviewed status blocks should not be removed casually one section at a time. - Reviewed status blocks should be recorded in normalization reports if removed from public-facing prose. - A section can be accepted for promoted draft assembly without being final-publication-ready.

Header normalization

Header normalization is the process of turning review-oriented section headers into final-manual section headers.

Header normalization may include: - removing Candidate note blocks from public-facing copies; - preserving provenance in reports; - standardizing title and section metadata; - checking section order and table of contents alignment.

Safe wording: - Header normalization must not rewrite substantive prose without explicit review. - Header normalization must be systematic. - Header normalization should report every section changed. - Header normalization should preserve originals or backups. - Header normalization is not final publication by itself.

Substantive prose boundary

Header normalization should not become stealth editing.

Substantive prose includes: - conceptual explanations; - evidence doctrine; - command behavior claims; - canary language; - examples; - scope boundaries; - future metadata feeder descriptions.

Safe wording: - Header normalization may clean review headers. - Header normalization should not change substantive prose without explicit authorization. - Substantive repairs should go through a repair/review path, not

a normalization pass. - If a header-normalization script changes body prose, that is a failure unless explicitly authorized.

Path repair and canonical path verification

Promoted drafts should verify canonical paths.

Path repair and canonical path verification should check: - section id; - expected filename; - actual file existence; - readable content; - hash match with source candidate where expected; - legacy or non-canonical path leftovers; - duplicate section files.

Safe wording: - Path and slug verification must use canonical section ids and inspectable files. - A status report is not enough if the file cannot be opened. - Path repair should preserve evidence and avoid deletion unless separately authorized. - Canonical path verification should be part of publication-readiness review.

Slugs and section ids

A slug or filename is not just formatting. It controls whether humans and scripts can find the section.

Safe wording: - Slugs should be derived from reviewed section ids. - Slug changes should be reported. - Non-canonical slugs should be flagged before publication. - Slug repair should preserve old evidence until cleanup is explicitly authorized.

Section count

Section count is a basic workspace integrity check.

A promoted draft review should report: - total section files; - expected required sections; - missing required sections; - extra or duplicate sections; - sections with candidate headers; - sections with reviewed-status blocks; - sections needing header normalization.

Safe wording: - Section count is a workspace health check. - Section count does not prove prose quality. - Section count should be combined with inspectable file checks.

Section order

Section order matters because the manual should read coherently.

A section-order review should check: - onboarding and orientation before deep internals; - data model before navigation; - navigation before indexing; - indexing before expressions where appropriate; - command surface before command reference; - evidence doctrine before publication readiness; - appendices or reference sections after concepts.

Safe wording: - Section order should be explicit. - Section order should be reviewable. - Section order should not be inferred only from directory listing order. - A future table of contents or manifest should own final order.

Table of contents

A table of contents is the public navigation surface for the manual.

Publication readiness should eventually verify: - every TOC entry has a section file; - every required section file appears in the TOC or is intentionally excluded; - titles match; - order is intentional; - section ids and slugs are stable; - no draft-only sections appear accidentally; - no evidence-only reports appear as public manual chapters.

Safe wording: - Table of contents checks belong to publication readiness. - TOC readiness is not the same as final publication. - TOC generation should not delete source sections. - TOC generation should preserve auditability.

Final publication boundary

Final publication is separate from promoted draft workspace assembly.

Safe wording: - Final publication requires explicit authorization. - Promotion into a draft workspace is not final publication. - Header normalization is not final publication. - TOC readiness is not final publication. - Publication readiness may recommend repairs, but it should not silently publish or normalize.

Generated command pages and evidence artifacts

Generated command pages and evidence artifacts remain protected.

Publication-readiness work must not delete: - generated command pages; - HELP evidence; - metadata reports; - CMDHELPCHK reports; - review packets; - PIP reports; - source evidence reports; - canary reports; - promoted draft history.

Safe wording: - Generated command pages and evidence artifacts must not be deleted during publication-readiness work. - Cleanup recommendations may be reported. - Deletion or mutation requires a separate explicit authorization path.

HELP, CMDHELPCHK, and metadata boundaries

Publication-readiness review may read HELP, CMDHELPCHK, and metadata evidence.

It must not mutate: - HELP; - META; - CMDHELPCHK; - catalogs; - source; - runtime data; - production SelfDoc metadata.

Safe wording: - HELP explains. - CMDHELPCHK validates. - Metadata organizes. - Manualgen publication-readiness review remains report-only. - No mutation occurs without explicit authorization.

Report-only publication readiness

Publication readiness should start as report-only.

Report-only readiness can produce: - section inventory; - candidate header inventory; - reviewed status inventory; - canonical path check; - slug check; - table of contents check; - section order check; - review packet inventory; - unresolved canary inventory; - publication blocker list; - recommended repairs.

Safe wording: - Report-only readiness is safe by default. - Report-only readiness can recommend repair. - Report-only readiness must not silently normalize, publish, or delete evidence.

Human review

Human review remains the decision point.

Human review should decide: - HOLD; - REPAIR; - ACCEPT_FOR_PROMOTION; - READY_FOR_HEADER_NORMALIZ - READY_FOR_PUBLICATION_REVIEW; - NOT_READY_FOR_PUBLICATION.

Safe wording: - Human review should inspect prose, not only reports. - Human review should be recorded. - Authorization should follow inspection. - Publication should not be implied by prior section acceptance.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- candidate note headers not normalized one section at a time
- reviewed candidate status blocks systematic normalization
- header normalization no substantive prose rewrite
- final publication separate from promoted draft assembly
- canonical path slug verification inspectable files
- table of contents section order actual files
- review packets preferred human inspection before authorization
- generated command pages evidence artifacts no deletion
- help meta cmdhelpchk source runtime selfdoc no mutation
- publication readiness recommends not silently publishes

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- promoted draft workspace review is distinct from final publication;
- candidate note headers are not normalized casually one section at a time;
- reviewed candidate status blocks are handled systematically;
- header normalization does not rewrite substantive prose;
- canonical path and slug verification use inspectable files;
- section count, section order, and table of contents checks are represented;
- review packets are preserved as human-inspection surfaces;
- generated command pages and evidence artifacts are protected from deletion;
- HELP, META, CMDHELPCHK, source, runtime data, and production SelfDoc metadata are not mutated;
- publication readiness may recommend repairs but does not silently publish or normalize.

Recommended required tokens for later PIP-003: - promoted draft - review packet - inspection packet - human review - candidate note - reviewed candidate status - header normalization - publication readiness - final publication - table of contents - section order - section count - path repair - canonical path - slug - generated command pages - no deletion - HELP - CMDHELPCHK - metadata - SelfDoc - manualgen - PIP - report-only - no mutation

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no final publication
- no header normalization
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Runtime Evidence, Source Verification, and Canary Closure

Pippets used: - PIP-001 Target Selection - MDO-180 Target Selection - MDO-181 Draft Fill

Evidence boundary: - Runtime proves behavior. - Source defines implementation and subsystem ownership. - HELP explains. - Metadata organizes. - CMDHELPCHK validates. - SelfDoc preserves provenance. - Manualgen assembles.

Slow-lane warning: - This section touches runtime proof, source verification, canary lifecycle, legacy evidence cautions, report-only boundaries, and no-mutation safety. - Do not send this directly to generic PIP-003. - Run a slow-lane runtime/source/canary review first.

Evidence tokens under review: - runtime - source - SOURCE - SOURCE_FACT - SelfDoc - manualgen - PIP - canary - canary ledger - smoke test - shakedown - regression - proof - verification - evidence - build - release - HELP - CMDHELPCHK - metadata - crosswalk - report-only - no mutation

Draft notes: - This is conservative manual prose for slow-lane evidence review. - Runtime proof should include concrete commands, output, build context, and date where possible. - Source verification should identify implementation ownership and should not be demoted to sidecar evidence. - Canary rows remain visible until closed with evidence. - Closing a canary requires current evidence, not optimism or old design intent. - Legacy documents may be useful context but should not be treated as current fact without verification. - Manualgen and SelfDoc remain report-only unless explicitly authorized. - HELP, CMDHELPCHK, and metadata can guide review but cannot close

runtime or source canaries by themselves. - Evidence packages should not mutate HELP, META, CMDHELPCHK, catalogs, source, or runtime data during manual draft assembly.

Purpose of this section

This section explains how the Developer Manual should use runtime evidence, source verification, and canary closure.

It follows the HELP, Metadata, CMDHELPCHK, and Manualgen Alignment section because that section established the authority doctrine. This section operationalizes that doctrine into daily engineering practice: how runtime evidence is captured, how source verification is attached, how canaries remain visible, and how canaries are closed only with evidence.

The goal is not to turn every manual claim into a test case. The goal is to prevent unsupported claims from becoming permanent prose and to make sure important uncertainties remain visible until they are resolved.

Evidence doctrine in practice

The working doctrine remains:

- Runtime proves behavior.
- Source defines implementation and subsystem ownership.
- HELP explains.
- Metadata organizes.
- CMDHELPCHK validates.
- SelfDoc preserves provenance.
- Manualgen assembles.

In practice, this means a claim may need several evidence lanes before it is safe for final manual prose.

Examples: - A command behavior claim needs runtime evidence. - An implementation ownership claim needs source verification. - A HELP wording claim needs HELP evidence. - A metadata alignment claim needs metadata evidence. - A consistency claim may need CMDHELPCHK evidence. - A provenance claim needs SelfDoc or manualgen run evidence.

Runtime evidence

Runtime evidence proves observed behavior.

Useful runtime evidence includes: - exact commands that were run; - exact output or relevant output excerpts; - build configuration; - release/debug mode; - dataset or workspace used; - command path or script path; - date and session context; - pass/fail result; - any unexpected warnings or canary behavior.

Safe wording: - Runtime evidence proves what was observed in that run. - Runtime evidence should be labeled with enough context to reproduce or understand it. - Runtime evidence does not automatically define implementation ownership. - Runtime evidence can close behavior canaries when the observed behavior is current and relevant.

Examples of runtime evidence lanes: - smoke test; - shakedown; - regression run; - build output; - release verification; - manual command transcript; - script output; - comparison before and after a repair.

Source verification

Source verification attaches implementation and ownership evidence.

Useful source verification includes: - source file path; - header path; - function or class name; - source-comment contract; - source-miner report; - SOURCE_FACT row; - SelfDoc source-comment evidence; - handler or parser ownership note; - subsystem boundary note.

Safe wording: - Source defines implementation and subsystem ownership. - Source verification should identify the relevant file or owner where possible. - Source may verify why runtime behavior occurs. - Source is not merely a sidecar; it is authority for implementation structure.

Source verification is especially important for claims about: - parser dispatch; - command handlers; - storage ownership; - relation traversal ownership; - expression evaluation ownership; - memo payload lifecycle; - message emission; - backend boundaries; - generated/help/metadata alignment.

Runtime and source together

Runtime and source answer different questions.

Runtime answers: - What happened in this run? - What behavior can be observed? - Did the smoke/shakedown/regression pass?

Source answers: - Which subsystem owns the behavior? - Which implementation path handles it? - Is the observed behavior intentional, scaffold leakage, compatibility behavior, or a bug?

Safe wording: - Runtime proves behavior. - Source defines ownership. - A strong manual claim often needs both. - HELP, metadata, and CMDHELPCHK can guide the search, but they do not replace runtime/source evidence.

Canary lifecycle

A canary is a visible uncertainty, risk, or proof-sensitive behavior that must not disappear silently.

A canary can be: - open; - deferred; - narrowed; - reproduced; - partially closed; - closed with evidence; - superseded by a better canary; - converted into an issue or work item.

A canary should not be: - hidden because it is inconvenient; - closed because the expected behavior seems obvious; - closed because old design intent says it should work; - closed using HELP, metadata, or CMDHELPCHK alone when runtime/source proof is required.

Opening a canary

A canary should be opened when evidence shows a behavior, mismatch, ambiguity, or risk that could affect manual accuracy or project correctness.

Good canary records include: - short name; - observed behavior; - evidence source; - date or session; - affected subsystem; - proof needed for closure; - current status; - next recommended action.

Examples: - SET ORDER active tag reporting. - x64 workspace ERSATZ load reporting. - MIN/MAX command/function ambiguity. - AGGS internal family owner exposure. - memo backend attach on workspace open. - generated command page duplicate or slug collision.

Keeping canaries visible

Canaries remain visible until closed with evidence.

Visibility matters because older project documents, generated reports, and runtime sessions can overlap. A canary keeps the manual honest by saying: this claim is not ready for final prose, or this behavior needs proof-sensitive wording.

Safe wording: - Canary rows remain visible until closed with evidence. - Canaries can be deferred, but deferred does not mean resolved. - Canaries can be narrowed when evidence shows only part of the risk remains. - Canaries should be carried forward in review reports until closed, superseded, or explicitly transferred.

Closing a canary

Closing a canary requires current evidence.

Closure evidence may include: - runtime run showing the behavior; - source verification showing the implementation path; - regression test output; - smoke test result; - CMDHELPCHK report when the canary is about HELP/catalog consistency; - metadata readback when the canary is about metadata coverage; - manualgen report when the canary is about assembly output; - explicit human decision when the canary is a policy or surface decision.

Safe wording: - Closing a canary requires evidence. - The closure record should name the evidence. - The closure record should preserve enough context to audit the decision later. - Old design intent is not closure evidence by itself.

Legacy documents

Legacy documents are useful context.

They may contain: - old design intent; - old architecture notes; - historical problems; - early terminology; - superseded assumptions; - useful examples; - project memory.

But legacy documents should not be treated as current fact without verification.

Safe wording: - Old documents remember. - Recent summaries steer. - SelfDoc verifies. - Runtime proves. - Source defines. - Manuals explain.

Legacy evidence should be labeled when it is used: - historical context; - design intent; - superseded note; - unverified claim; - candidate canary; - verified current behavior.

Smoke tests, shakedown, regressions, builds, and releases

Different runtime evidence lanes serve different purposes.

Smoke test: - quick proof that a feature or path starts and behaves basically as expected.

Shakedown: - broader exploratory runtime evidence, often with manual commands and transcript notes.

Regression: - evidence that a previously fixed or expected behavior still works.

Build: - evidence that code compiles and links in a given configuration.

Release: - evidence attached to a release-ready state, usually after build and smoke checks.

Safe wording: - Build success is not behavior proof. - Smoke success is limited behavior proof. - Shakedown is useful runtime evidence but should be scoped. - Regression evidence is strong for previously known behavior. - Release evidence should identify build configuration and major checks.

REGRESSION and TEST as proof launchers

REGRESSION and TEST make repeatable proof entry points discoverable, but neither command makes a run self-proving. Retain the exact command, selected script or suite, build identity, transcript, and result counts when using either launcher as evidence.

REGRESSION REGRESSION [USAGE|LIST|SHOW <name>|RUN <name>|<name>|ALL] selects from a curated set of stable regression and shakedown scripts. LIST and SHOW expose the curated entries; RUN <name> and the compact <name> form launch one entry; ALL launches the defined ordered set. The command delegates execution to DOTSCRIPT, so it is a catalogued launcher rather than a separate test engine. Developer reproduction canaries that are not in the curated set remain outside ALL.

TEST TEST <scriptfile> [<logfile>] [VERBOSE] runs a specified test script through the shell test harness. The harness resolves the script path, handles supported inline comments and continued logical commands, executes those commands, and reports processed and error counts. A supplied logfile may be created or truncated. The script controls the resulting side effects, so TEST must not be classified as read-only merely because it is used for verification.

Evidence boundary Launcher availability proves only that the entry surface exists. A proof claim still requires the retained run inputs and outcome. Cross-reference the Scripting and Control Flow section for script semantics and Runtime Operation, Invocation, and Automation for entry-path behavior.

HELP, CMDHELPCHK, and metadata in evidence practice

HELP, CMDHELPCHK, and metadata are powerful review guides.

They can: - identify expected command vocabulary; - expose missing help or command topics; - organize arguments, messages, variants, and functions; - detect catalog/help drift; - point to canaries; - guide runtime/source verification.

They cannot by themselves: - prove runtime behavior; - define implementation ownership; - close runtime canaries; - close source ownership canaries.

Safe wording: - HELP explains. - CMDHELPCHK validates. - Metadata organizes. - Runtime/source evidence closes runtime/source canaries.

SelfDoc and manualgen evidence

SelfDoc preserves provenance. Manualgen assembles.

SelfDoc evidence may include: - source-comment contract reports; - SOURCE_FACT rows; - source inventory; - classifier reports; - diagram reports; - canary ledgers; - provenance notes.

Manualgen evidence may include: - pipet run records; - target selection reports; - draft fill reports; - slow-lane reviews; - PIP-003 evidence gates; - PIP-004 reviewed candidates; - PIP-005 human decisions; - PIP-006 promotion patches; - promoted draft workspaces.

Safe wording: - SelfDoc and manualgen are evidence systems. - They default to report-only. - They must not mutate HELP, META, CMDHELPCHK, catalogs, source, or runtime data during manual draft assembly. - Production mutation requires explicit authorization.

Evidence crosswalks

Crosswalks connect evidence lanes without collapsing them.

Useful crosswalks include: - runtime transcript to canary row; - canary row to source file; - source file to SOURCE_FACT; - HELP topic to command identity; - command identity to SYSCMD; - function claim to SYSFUNC; - diagnostic claim to SYSMMSG; - argument claim to SYSARGS; - alias/variant claim to SYSENTVAR; - manual section to evidence tokens; - PIP report to promoted draft section.

Safe wording: - Crosswalks preserve traceability. - Crosswalks can be candidate, partial, or verified. - Crosswalks should not pretend a weak source is strong evidence. - Crosswalks help future metadata feeders absorb temporary evidence.

Future META alignment

This section should eventually align with evidence metadata.

Expected future feeders and evidence lanes: - SOURCE_FACT for source/comment provenance and implementation ownership evidence. - SRCFILE, SRCBLOCK, SRCLINE, SRCUSAGE, SRCCLASS, SRCDISP, SRCALIAS, and MEMO_LINES for SelfDoc source-comment evidence where available. - SYSCMD and SYSSUBCMD for command and subcommand identity tied to evidence. - SYSFUNC for function evidence and function-command bridge canaries. - SYSMMSG for diagnostic and message canaries. - SYSHELP for help alignment and curated help evidence. - SYSARGS for argument-shape evidence and validation surfaces. - SYSENTVAR for aliases, variants, and command/function entry variants. - Manualgen PIP reports for gate evidence and assembly provenance. - Canary ledger reports for open, deferred, reproduced, narrowed, and closed canary rows.

No-mutation safety

Evidence packages should not mutate production artifacts during manual draft assembly.

Default boundary: - no generated command page deletion; - no HELP mutation; - no META mutation; - no CMDHELPCHK mutation; - no catalog apply; - no source edits; - no runtime data mutation; - no production SelfDoc metadata promotion; - no final publication.

Report-only work is the default.

Slow-lane canary tracking names

The slow-lane review tracks these canaries by exact name. These names are review anchors, not final user-facing prose.

- runtime proof concrete commands output build context date
- source verification implementation ownership not sidecar
- canary rows visible until closed with evidence

- canary closure requires current evidence not optimism old design intent
- legacy documents context not current fact without verification
- manualgen selfdoc report-only unless authorized
- help cmdhelpchk metadata guide not close runtime source canaries
- build smoke shakedown regression labeled dated
- evidence packages no mutation
- canary deferred narrowed reproduced closed not disappear

These anchors preserve the canaries that the prose discusses in ordinary language. They should remain until the section is promoted through evidence review.

Canary non-disappearance boundary

This exact review anchor is intentionally retained for slow-lane evidence review:

- A canary may be deferred/narrowed/reproduced/closed, but should not disappear silently.

The anchor is not final user-facing prose by itself. It preserves the boundary already described in this section: canaries remain visible until closed, superseded, transferred, or explicitly deferred with evidence. ### Review notes before PIP-003

This is a slow-lane section. Before generic PIP-003 is allowed to create a reviewed-candidate path, an MDO slow-lane evidence review should check:

- all required tokens are represented or intentionally excluded;
- runtime proof concrete commands/output/build/date boundary is present;
- source verification/implementation ownership boundary is present;
- canary visibility and closure rules are present;
- old design intent and legacy documents are not treated as current fact without verification;
- HELP/CMDHELPCHK/metadata are review guides, not canary closure evidence by themselves;
- build, smoke, shakedown, regression, and release evidence are scoped correctly;
- SelfDoc/manualgen report-only and no-mutation boundaries are visible;
- future META/evidence feeders are preserved.

Recommended required tokens for later PIP-003: - runtime - source - SOURCE - SOURCE_FACT - SelfDoc - manualgen - PIP - canary - canary ledger - smoke test - shakedown - regression - proof - verification - evidence - build - release - HELP - CMDHELPCHK - metadata - crosswalk - report-only - no mutation

Boundary

- prose draft fill only
- slow-lane review still required
- no reviewed candidate generated
- no final prose promotion
- no generated command page deletion
- no HELP mutation
- no META mutation
- no CMDHELPCHK mutation
- no catalog apply
- no source edits
- no production SelfDoc metadata promotion

Diagrams

12 promoted diagrams bound from the attachment matrix (docs/manuals/developer/manualgen/reports/diagram-publicati each placed at its manual target and carrying its proof level.

Diagram	Source asset	Manual target	Proof
DIAG-SELFDCCORE-001	docs/manuals/assets/diagrams/DEV-19	DEV-19	generated-reviewed
DIAG-SDLCPARALLEL-002	docs/media/LabTalk_DotTalk_19_Systems_Storyboard_Deck/slideview.png	DEV-19	generated-reviewed
DIAG-SDLCURATION-003	labtalk/diagrams/LABTALK_DotTalk_19_Systems_Storyboard_Deck/slideview.png	DEV-19	generated-reviewed
DIAG-CMDAUTH-004	docs/manuals/assets/diagrams/DEV-05	DEV-05	generated-reviewed
DIAG-X64GEOM-005	docs/manuals/assets/diagrams/DEV-08	DEV-08	generated-reviewed
DIAG-X64SELFDESC-006	docs/manuals/assets/diagrams/DEV-08	DEV-08	generated-reviewed
DIAG-INDEXARCH-007	docs/manuals/assets/diagrams/DEV-09	DEV-09	generated-reviewed
DIAG-WORKAREA-008	docs/manuals/assets/diagrams/DEV-13	DEV-13	generated-reviewed
DIAG-ARCHLAYERS-009	docs/manuals/assets/diagrams/DEV-04	DEV-04	generated-reviewed
DIAG-LABCAMPUS-010	labtalk/diagrams/labtalkcampuscapv0s.svg	DEV-09	generated-reviewed
DIAG-LABPROOF-011	labtalk/diagrams/labtalkproofcapv0s.svg	DEV-09	generated-reviewed
DIAG-MDOFLOW-012	labtalk/diagrams/mdo_manualgen_publication_dataflowv01-mmd	DEV-09	generated-reviewed

Appendices

part pending generator (manualgen build-dry-run (section candidate))

Glossary

Review candidate — terms harvested from command/function names and section headings; definitions are pending maintainer review (**candidate** mode). Note: the DotTalk++ **GLOSSARY** command is documented in the Command Reference, distinct from this back-matter glossary.

Term	Definition
CDX	Compound index file (multiple ordered tags).
DBF	dBASE-family table file; x64base uses a 64-bit-widened variant (DBF_64).
FPT	Memo (variable-length field) file paired with a DBF; x64base uses FPT64.
LMDB	Lightning Memory-Mapped Database backend used for index/order acceleration.
MDO	Master Documentation Organizer — lane/section/promotion structure.
RECNO	Record number; x64base widens record addressing to 64-bit (RECNO64).
SelfDoc	The report-only documentation-generation subsystem.
WAL	Write-ahead log providing durability for buffered mutations.
ABS	<i>definition — review</i>
ACOS	<i>definition — review</i>
ALLTRIM	<i>definition — review</i>
APPEND	<i>definition — review</i>
AREA	<i>definition — review</i>
AREA51	<i>definition — review</i>
ASC	<i>definition — review</i>
ASCEND	<i>definition — review</i>
ASCII	<i>definition — review</i>
ASIN	<i>definition — review</i>
AT	<i>definition — review</i>
ATAN	<i>definition — review</i>
ATC	<i>definition — review</i>

Term	Definition
AUTODBF	<i>definition — review</i>
BETWEEN	<i>definition — review</i>
BIBLETALK	<i>definition — review</i>
BOTTOM	<i>definition — review</i>
BROWSE	<i>definition — review</i>
BROWSER	<i>definition — review</i>
BUFFERING	<i>definition — review</i>
BUILDLMDB	<i>definition — review</i>
CANARY	<i>definition — review</i>
CDOW	<i>definition — review</i>
CEILING	<i>definition — review</i>
CHR	<i>definition — review</i>
CHRISTMAS	<i>definition — review</i>
CHRTRAN	<i>definition — review</i>
CLOSE	<i>definition — review</i>
CMDREL	<i>definition — review</i>
CMONTH	<i>definition — review</i>
CNX	<i>definition — review</i>
COBOL	<i>definition — review</i>
CODASYL	<i>definition — review</i>
COMMANDSHELP	<i>definition — review</i>
COMMIT	<i>definition — review</i>
CONCAT	<i>definition — review</i>
CONTINUE	<i>definition — review</i>
COPY	<i>definition — review</i>
COS	<i>definition — review</i>
CREATE	<i>definition — review</i>
...	<i>184 further harvested terms pending review</i>

Index

A

- ABS (function)
- ACOS (function)
- AGGS boundary
- Aliases and entry variants
- Aliases and variants
- ALLTRIM
- ALLTRIM (command)
- ALLTRIM (function)
- APPEND
- APPEND (command)
- APPEND BLANK and APPEND_BLANK
- Appendices
- Appendix: Review and Deferred: SET-family
- AREA
- AREA (command)
- AREA51
- AREA51 (command)
- ASC
- ASC (command)
- ASC (function)
- ASCEND
- ASCEND (command)

- ASCII
- ASCII (command)
- ASIN (function)
- Assembly workflow versus truth authority
- AT
- AT (command)
- AT (function)
- ATAN (function)
- ATC
- ATC (command)
- ATC (function)
- Authority boundaries
- Authority model for command reference assembly
- AUTODBF
- AUTODBF (command)

B

- BETWEEN (function)
- BIBLETALK
- BIBLETALK (command)
- BOTTOM
- BOTTOM (command)
- Boundary
- BROWSE
- BROWSE (command)
- BROWSER
- BROWSER (command)
- BUFFERING
- BUFFERING (command)
- BUILDLMDB
- BUILDLMDB (command)

C

- CALC and CALCWRITE
- CANARY
- CANARY (command)
- Canary lifecycle
- Canary non-disappearance boundary
- Candidate note headers
- Canonical command identity
- Canonical commands
- CDOW (function)
- CDX
- CDX (command)
- CDX, CNX, and LMDB boundary
- CEILING (function)
- CHR
- CHR (command)
- CHR (function)
- CHRISTMAS
- CHRISTMAS (command)
- CHRTRAN (function)
- CLOSE
- CLOSE (command)
- Closing a canary
- CMDHELPCHK lane

- CMDHELPCHK role
- CMDREL
- CMDREL (command)
- CMONTH (function)
- CNX
- CNX (command)
- COBOL
- COBOL (command)
- CODASYL
- CODASYL (command)
- Colophon — build provenance
- Command and concept map
- Command Reference
- Command Reference Assembly, Aliases, and Generated Page Hygiene
- Command surface
- Command Surface, Dispatch, and Entry Variants
- Command-reference crosswalks
- Commands in this section
- COMMANDSHELP
- COMMANDSHELP (command)
- COMMIT
- COMMIT (command)
- Compatibility and bridge material
- CONCAT
- CONCAT (command)
- CONCAT (function)
- CONTINUE
- CONTINUE (command)
- COPY
- COPY (command)
- Core doctrine
- COS (function)
- COUNT and aggregate commands
- CREATE
- CREATE (command)
- Crosswalk discipline
- CTOD
- CTOD (command)
- CTOD (function)

D

- DATE
- DATE (command)
- DATE (function)
- DATEADD (function)
- DATEDIFF (function)
- DATETIME (function)
- DAY (function)
- DECISION
- DECISION (command)
- DELETE
- DELETE (command)
- Deleted-record filters
- DESCEND
- DESCEND (command)
- Diagrams

- DIR
- DIR (command)
- Direct aggregate verbs and AGGS
- DO
- DO (command)
- DOTHELP
- DOTHELP (command)
- DOTSCRIPT
- DOTSCRIPT (command)
- DOW (function)
- DRAWIO
- DRAWIO (command)
- DTOC
- DTOC (command)
- DTOC (function)
- DTOS (function)
- DUMP
- DUMP (command)
- Duplicate command rows

E

- EDIT
- EDIT (command)
- EDUCATIONAL_USE
- EDUCATIONAL_USE (command)
- ELSE
- ELSE (command)
- EMPTY (function)
- ENDIF
- ENDIF (command)
- ENDLOOP
- ENDLOOP (command)
- ENDSCAN
- ENDSCAN (command)
- ENDUNTIL
- ENDUNTIL (command)
- ENDWHILE
- ENDWHILE (command)
- ERASE
- ERASE (command)
- ERP
- ERP (command)
- Error / Message Catalog
- Error and null behavior
- ERSATZ
- ERSATZ (command)
- ERSATZ and browser caution
- EVAL
- EVAL (command)
- EVALUATE
- EVALUATE (command)
- Evidence crosswalks
- Evidence doctrine in practice
- Evidence lanes
- EXAMPLE
- EXAMPLE (command)

- EXP (function)
- EXPFUNCS
- EXPFUNCS (command)
- EXPORT
- EXPORT (command)
- EXPORTSQL
- EXPORTSQL (command)
- EXPRESSION
- EXPRESSION (command)
- Expression evaluation surfaces
- Expressions, Querying, and Aggregates

F

- Final publication boundary
- FIND
- FIND (command)
- FLOOR (function)
- FOXHELP
- FOXHELP (command)
- FOXPRO
- FOXPRO (command)
- FOXSTANDARD
- FOXSTANDARD (command)
- FOXTALK
- FOXTALK (command)
- Function bridge behavior
- Function command-line bridge
- Function Reference
- Future META alignment

G

- Generated command pages
- Generated command pages and evidence artifacts
- GLOSSARY
- Glossary
- GLOSSARY (command)
- GO
- GO (command)
- GOMONTH (function)
- GOTO
- GOTO (command)
- GPS
- GPS (command)

H

- Header normalization
- HELP and CMDHELPCHK
- HELP FUNCTIONS and FUNCTION help
- HELP lane
- HELP rows as diagnostic evidence
- HELP, CMDHELPCHK, and metadata boundaries
- HELP, CMDHELPCHK, and metadata in evidence practice
- HELP, Metadata, CMDHELPCHK, and Manualgen Alignment
- Human review

I

- IDX
- IDX (command)
- IF
- IF (command)
- IMAGE
- IMAGE (command)
- IMPORT
- IMPORT (command)
- IMPORTSQL
- IMPORTSQL (command)
- INDEX (command)
- Indexing vocabulary
- Indexing, Tags, Relations, and Views
- INIT
- INIT (command)
- Inspectable files
- INT (function)
- Internal owner and public surface
- INTRO
- INTRO (command)

K

- Keeping canaries visible
- Known canaries

L

- LEFT
- LEFT (command)
- LEFT (function)
- Legacy documents
- LEN
- LEN (command)
- LEN (function)
- LIKE (function)
- LIST
- LIST (command)
- LMDB
- LMDB (command)
- LMDB_UTIL
- LMDB_UTIL (command)
- LMDBDUMP
- LMDBDUMP (command)
- LOAD guard
- LOCATE
- LOCATE (command)
- LOCATE, CONTINUE, and SCAN
- LOCK
- LOCK (command)
- LOG (function)
- LOG10 (function)
- Logical order and active order
- LOOP
- LOOP (command)
- LOOP_BUFFER
- LOOP_BUFFER (command)
- LOOPS

- LOOPS (command)
- LOWER
- LOWER (command)
- LOWER (function)
- LTRIM
- LTRIM (command)
- LTRIM (function)

M

- Manualgen lane
- MAX (function)
- MCC
- MCC (command)
- Message catalog and HELP alignment
- Message vocabulary
- Messages, Errors, and Diagnostics
- Metadata feeders
- Metadata lane
- MIN (function)
- MOD (function)
- MODEL
- MODEL (command)
- MONTH (function)

N

- NAVIGATION
- NAVIGATION (command)
- No-active-table and not-found messages
- No-delete and no-mutation safety
- No-mutation safety
- NORMALIZE
- NORMALIZE (command)
- Notes for future prose pass
- NOW (function)

O

- Open index architecture note
- Opening a canary
- ORDER
- ORDER (command)
- Ownership reminder

P

- PACK
- PACK (command)
- PADC
- PADC (command)
- PADL
- PADL (command)
- PADR
- PADR (command)
- Parser dispatch and handlers
- Parser warnings and syntax diagnostics
- Path repair and canonical path verification
- PREDHELP
- PREDHELP (command)

- PREDICATE
- PREDICATE (command)
- PREDICATES
- PREDICATES (command)
- Predicates, WHERE, and FOR
- Preface — how to read this manual
- PROJECTION
- PROJECTION (command)
- PROJECTS
- PROJECTS (command)
- Promoted Draft Review, Header Normalization, and Publication Readiness
- Promoted draft workspace
- PROPER
- PROPER (command)
- Provenance & attestation
- PSHELL
- PSHELL (command)
- Publication readiness
- Purpose of this section

R

- RAND (function)
- RAT (function)
- REBUILD
- REBUILD (command)
- RECALL
- RECALL (command)
- Record views and memo fields
- REGRESSION and TEST as proof launchers
- REINDEX
- REINDEX (command)
- Reindexing and rebuild behavior
- REL
- REL (command)
- REL_ENUM
- REL_ENUM (command)
- REL_REFRESH
- REL_REFRESH (command)
- RELATION
- RELATION (command)
- RELATIONS
- RELATIONS (command)
- Relations and relation traversal
- REPLACE
- REPLACE (command)
- REPLICATE
- REPLICATE (command)
- REPLICATE (function)
- Report-only publication readiness
- RETRO
- RETRO (command)
- Review notes before candidate generation
- Review notes before PIP-003
- Review packets and inspection packets
- Reviewed candidate status blocks
- RIGHT

- RIGHT (command)
- RIGHT (function)
- ROLLBACK
- ROLLBACK (command)
- ROUND (function)
- RTRIM
- RTRIM (command)
- RTRIM (function)
- RULE
- RULE (command)
- RUN
- RUN (command)
- Runtime and source together
- Runtime diagnostic examples
- Runtime evidence
- Runtime Evidence, Source Verification, and Canary Closure
- Runtime lane

S

- Safety boundaries
- Scalar functions versus aggregate commands
- SCAN
- SCAN (command)
- SCAN_BUFFER
- SCAN_BUFFER (command)
- Schema and DDL
- SCHEMAS
- SCHEMAS (command)
- SCRIPT
- SCRIPT (command)
- SCX
- SCX (command)
- SECHO
- SECHO (command)
- SECONDS (function)
- Section count
- Section order
- SECURITY
- SECURITY (command)
- SEEK
- SEEK (command)
- SEEK and FIND active-order boundary
- SELECT
- SELECT (command)
- SelfDoc and manualgen evidence
- SelfDoc lane
- SET family boundary
- SET-family boundary and canonicalization canary
- SET-family canonicalization
- Severity vocabulary
- SFTP
- SFTP (command)
- SHARED_MSG caution
- HELLO
- HELLO (command)
- SHOWINI

- SHOWINI (command)
- SHUTDOWN
- SHUTDOWN (command)
- SIN (function)
- SIX
- SIX (command)
- SKIP
- SKIP (command)
- Slow-lane canary tracking names
- Slug collisions
- Slugs and section ids
- SMARTLIST
- SMARTLIST (command)
- Smoke tests, shakedowns, regressions, builds, and releases
- SOUNDEX (function)
- Source lane
- Source verification
- Source-lane rule for this section
- SPACE
- SPACE (command)
- SPACE (function)
- SQL
- SQL (command)
- SQLERASE
- SQLERASE (command)
- SQLHELP
- SQLHELP (command)
- SQLITE
- SQLITE (command)
- SQLSEL
- SQLSEL (command)
- SQLVER
- SQLVER (command)
- SQRT (function)
- STATE
- STATE (command)
- STR
- STR (command)
- STR (function)
- STRTRAN (function)
- STRUCT
- STRUCT (command)
- STU_REPEAT
- STU_REPEAT (command)
- STU_UPPER
- STU_UPPER (command)
- STUDENTECHO
- STUDENTECHO (command)
- STUDENTHELLO
- STUDENTHELLO (command)
- STUFF
- STUFF (command)
- Subcommands and command families
- Substantive prose boundary
- SUBSTR
- SUBSTR (command)

- SUBSTR (function)
- SYSMSG and SYSTEM_MESSAGES

T

- TABLE_BUFFER
- TABLE_BUFFER (command)
- Tables, Records, and Data Model
- Tables, records, and fields
- Tags and tag availability
- TAN (function)
- Temporary evidence lanes
- TEXT
- TEXT (command)
- TIME
- TIME (command)
- TIME (function)
- TODAY (function)
- TOP
- TOP (command)
- TRANSFORM (function)
- TRIM
- TRIM (command)
- TUPEXPORT
- TUPEXPORT (command)
- TUPLE
- TUPLE (command)
- TUPLEDELTA
- TUPLEDELTA (command)
- TUPTALK
- TUPTALK (command)
- TUPVALIDATE
- TUPVALIDATE (command)
- TURBOPACK
- TURBOPACK (command)
- TVISION
- TVISION (command)
- Typed message catalog direction

U

- UNDELETE
- UNDELETE (command)
- UNLOCK
- UNLOCK (command)
- UNTIL
- UNTIL (command)
- UNTIL_BUFFER
- UNTIL_BUFFER (command)
- UPDATE
- UPDATE (command)
- UPPER
- UPPER (command)
- UPPER (function)
- USE
- USE (command)

V

- VAL
- VAL (command)
- VAL (function)
- VALIDATE
- VALIDATE (command)
- Views and projection boundary

W

- WEB
- WEB (command)
- What this section should not do yet
- WHILE
- WHILE (command)
- WHILE_BUFFER
- WHILE_BUFFER (command)
- Work areas as context, not the main topic
- WORKSPACE
- WORKSPACE (command)
- WSREPORT
- WSREPORT (command)

Y

- YEAR (function)

Z

- ZAP
- ZAP (command)
- ZIP
- ZIP (command)

Colophon — build provenance

This manual assembled itself. The record below is emitted by the assembler so the self-documentation claim is auditable end to end.

Assembler	manual-assembler/0.1.0 (MANUAL-ASSEMBLY M3)
Manifest	tools/manualgen/manual_assembly_manifest.yaml (dottalk.manual.assembly_manifest.v1)
Parts assembled	22 (of 22 declared)
Greenfield parts generated	8
Public source snapshot	8ee746dee
Python (build)	3.10.12 (target 3.12)
Build timestamp (UTC)	2026-07-20T19:17:26Z
Machine	Alienware m16 R2 — MAINTAINER_ATTESTED
Accepted reader baseline	EA2E12A9D3E1