

README

BIBLOS LOGOU

Orthodox Exegetical Commentary System

A comprehensive database-driven system for generating scholarly Orthodox Christian biblical commentary, implementing the Stratified Foundation System and Fourfold Sense methodology.

[DOC] What's In The System

Currently Implemented

- * [x] **Fourfold Sense Analysis** (Literal, Allegorical, Tropological, Anagogical)
- * [x] **Nine-Matrix Verse Processing** (Emotional valence, theological weight, etc.)
- * [x] **Stratified Foundation System** (Seven layers of narrative depth)
- * [x] **Orbital Resonance Motif Tracking** (10 primary motifs)
- * [x] **Thread Density Management** (18-22 target bounds)
- * [x] **Patristic Integration** (Church Fathers commentary)
- * [x] **Typological Network Building** (Type-antitype relationships)
- * [x] **Cross-Reference Analysis** (Scripture interconnections)
- * [x] **AI-Enhanced Processing** (OpenAI, Claude, or local templates)
- * [x] **Batch Orchestration** (Parallel processing with checkpoints)
- * [x] **Validation Suite** (Invisibility checks, density validation)
- * [x] **Analytics Dashboard** (Processing metrics, motif status)

Not Yet Implemented

- * [] HTML output generation
 - * [] LaTeX/print-ready output
 - * [] Full 73-book verse population
 - * [] Complete patristic source library
 - * [] Web interface
-

[BUILD]■ System Architecture

```
BIBLOS LOGOU/
--- main.py # Central CLI entry point
--- requirements.txt # Python dependencies
--- .env.example # Environment configuration template
--- bible_refinement_db.sql # PostgreSQL database schema
```

```

|
+-- config/
|   +-- settings.py # Central configuration management
|
+-- scripts/
|   +-- __init__.py
|   +-- database.py # Database connection & repositories
|   +-- ingestion.py # Data ingestion pipeline
|   +-- processing.py # Verse processing pipeline
|   +-- output_generator.py # Output generation (MD, JSON)
|   +-- analytics.py # Analytics and reporting
|   +-- orchestration.py # Batch processing orchestration
|   +-- validation.py # Quality assurance validation
|   +-- integrity.py # Data integrity checks
|   +-- narrative_engine.py # Narrative generation engine
|
+-- tools/
|   +-- __init__.py
|   +-- ai_integration.py # AI provider integration (OpenAI, Claude, local)
|   +-- bible_api.py # Bible API integration
|   +-- cross_references.py # Cross-reference analysis & typology
|   +-- patristic_integration.py # Patristic sources & catena generation
|   +-- sensory_vocabulary.py # Sensory vocabulary codex management
|
+-- data/ # Precomputed data modules
|   +-- __init__.py
|   +-- character.voices.py # Character voice profiles
|   +-- cross_references.py # Cross-reference data
|   +-- liturgical_calendar.py # Liturgical calendar mappings
|   +-- morphology.py # Hebrew/Greek morphology data
|   +-- narrative_order.py # Hermeneutical event ordering
|   +-- nine_matrix.py # Nine-matrix calculation data
|   +-- offline_bible.py # Offline Bible text
|   +-- orthodox_study_bible.py # OSB annotations
|   +-- patristic_data.py # Patristic commentary data
|   +-- precomputed.py # Precomputed verse analyses
|   +-- sensory_vocabulary.py # Sensory vocabulary codex
|   +-- unified.py # Unified data access layer
|
+-- docs/
|   +-- IMPLEMENTATION_GUIDE.md # Detailed implementation guide
|   +-- WORKFLOW_GUIDE.md # End-to-end workflow guide
|
+-- output/ # Generated output files
+-- logs/ # Log files

```

[BOOK] Important Reference Files

These files contain the theological and methodological foundation for the system:

File	Description
MASTER_PLAN.md	The definitive master plan containing the complete
BIBLOS_LOGOU_EXPANDED METHODOLOGY.md	Expanded methodology for verbal perfection and sys
Hermeneutical.txt	Tonal ordering considerations for event sequencing

Stratified.txt	The Stratified Foundation System architecture defi
Biblical_Events_Complete_Orthodox_Canon.txt	Complete events list for the Eastern Orthodox canon
REFINED MASTER OUTLINE.md	The refined outline for Genesis 5 and surrounding

[ROCKET] Quick Start

1. Prerequisites

- * Python 3.9+
- * PostgreSQL 14+
- * (Optional) OpenAI or Anthropic API key for AI-enhanced generation

2. Installation

```
# Clone the repository
git clone https://github.com/your-repo/biblos-logou.git
cd biblos-logou

# Create virtual environment
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Copy environment configuration
cp .env.example .env
# Edit .env with your database credentials
```

3. Database Setup

```
# Create the database
createdb biblos_logou

# Initialize schema and data
python main.py init --schema bible_refinement_db.sql --all
```

4. Basic Usage

```
# Check system status
python main.py status

# Process verses
python main.py process --batch 100

# Export results
python main.py export --dashboard
python main.py export --book "Genesis" --format markdown
```

[BOOK] Core Concepts

Fourfold Sense of Scripture

The system implements the traditional fourfold hermeneutical method:

Sense	Weight	Description
Literal	30%	Historical-grammatical meaning
Allegorical	25%	Christological-typological significance
Tropological	25%	Moral-formational application
Anagogical	20%	Eschatological-heavenly meaning

Stratified Foundation System

Seven layers of narrative depth:

- * **Layer One** (Surface Adjacency): 0-50 pages
- * **Layer Two** (Near Foundation): 50-200 pages
- * **Layer Three** (Mid-Foundation): 200-500 pages
- * **Layer Four** (Deep Foundation): 500-1200 pages
- * **Layer Five** (Bedrock Foundation): 1200-2500 pages
- * **Layer Six** (Structural Undercurrent): Continuous
- * **Layer Seven** (Theological Bedrock): Eternal

Nine-Matrix Elements

Each verse is analyzed across nine dimensions:

- * Emotional Valence (0.0-1.0)
- * Theological Weight (0.0-1.0)
- * Narrative Function
- * Sensory Intensity (0.0-1.0)
- * Grammatical Complexity (0.0-1.0)
- * Lexical Rarity (0.0-1.0)
- * Breath Rhythm
- * Register Baseline
- * Tonal Weight

Orbital Resonance

Motifs follow harmonic trajectories:

- * **Planting** (95% intensity)
- * **Reinforcement at 1/2** (90%)
- * **Reinforcement at 5/6** (60%)
- * **Reinforcement at 15/16** (30%)
- * **Convergence** (100%)

Thread Density

Target: **18-22** active thread-points per 50-page span

[TOOL] CLI Commands

`init` - Initialize Database

```
python main.py init --schema bible_refinement_db.sql --all  
python main.py init --motifs # Initialize motifs only
```

`ingest` - Ingest Data

```
python main.py ingest --verses data/verses.txt
```

`process` - Process Verses

```
python main.py process --batch 100 # Process batch  
python main.py process --continuous # Process all  
python main.py process --verse-id 1234 # Process specific verse
```

`export` - Generate Output

```
python main.py export --dashboard # Progress dashboard  
python main.py export --book "Genesis" # Export single book  
python main.py export --all --format both # Export everything
```

`status` - System Status

```
python main.py status
```

`fetch` - Fetch Verse Text

```
python main.py fetch --verse "Genesis 1:1"  
python main.py fetch --populate --limit 100
```

`validate` - Run Validation Checks

```
python main.py validate --full # Full validation suite  
python main.py validate --verse-id 1234 # Validate specific verse  
python main.py validate --density-page 500 # Check density at page
```

`analytics` - Generate Analytics

```
python main.py analytics --report # Full analytics report  
python main.py analytics --processing # Processing velocity  
python main.py analytics --motifs # Motif status overview
```

`orchestrate` - Batch Orchestration

```
python main.py orchestrate --plan sequential # View processing plan  
python main.py orchestrate --execute_by_category # Execute plan  
python main.py orchestrate --list-checkpoints # List checkpoints  
python main.py orchestrate --run # Run batch processing
```

`patristic` - Patristic Integration

```
python main.py patristic --list-fathers # List Church Fathers  
python main.py patristic --father "Augustine" # Father info  
python main.py patristic --verse "Gen 1:1" # Commentary for verse  
python main.py patristic --catena "John 1:1" # Generate catena
```

`crossref` - Cross-Reference Operations

```
python main.py crossref --init-typology # Initialize typological pairs  
python main.py crossref --analyze "Gen 22:2" # Analyze references  
python main.py crossref --suggest "Exodus 12:13" # Suggest references  
python main.py crossref --stats # Network statistics
```

[GEAR]■ Processing Pipeline

Verses progress through these stages:

RAW -> PARSED -> ANALYZED -> STRATIFIED -> FLESHED_OUT -> TONALLY_ADJUSTED -> REFINED -> VERIFIED

Stage	Description
Raw	Initial state after ingestion
Parsed	Text extracted and normalized
Analyzed	Nine-matrix elements calculated
Stratified	Foundation layers assigned
Fleshed Out	Fourfold senses expanded
Tonally Adjusted	Hermeneutical ordering applied
Refined	Final polish complete
Verified	Passed invisibility checks

[DB]■ Database Schema

Core Tables

Table	Description
`canonical_books`	73 Orthodox canonical books
`verses`	37,454+ verses with fourfold senses
`events`	Biblical events for tonal arrangement
`motifs`	Orbital and standard motifs
`motif_activations`	Individual motif appearances
`typological_correspondences`	Type-antitype relationships
`sensory_vocabulary`	Sensory vocabulary codex
`patristic_sources`	Church Fathers commentary
`hermeneutical_principles`	Tonal ordering principles
`cross_references`	Scripture cross-references

Key Views

- * `vw_next_processing_batch` - Verses ready for processing
- * `vw_current_density` - Thread density status
- * `vw_approaching_convergences` - Motifs nearing convergence
- * `vw_fourfold_completion` - Sense completion status

[BOT] AI Integration

The system supports multiple AI providers:

OpenAI

```
# In .env
AI_PROVIDER=openai
AI_API_KEY=sk-your-key
AI_MODEL=gpt-4
```

Claude (Anthropic)

```
# In .env
AI_PROVIDER=claude
AI_API_KEY=sk-ant-your-key
AI_MODEL=claude-3-sonnet-20240229
```

Local (Template-based)

```
# In .env (default, no API needed)
AI_PROVIDER=local
```

[EXPORT] Output Formats

Markdown

Full commentary with:

- * Fourfold sense analysis
- * Nine-matrix elements
- * Tonal characteristics
- * Table of contents

JSON

Structured data export for:

- * Integration with other systems
- * API consumption
- * Data analysis

Progress Dashboard

Real-time status including:

- * Completion percentage
 - * Book-by-book progress
 - * Processing statistics
-

[TARGET] Primary Motifs

The system tracks 10 primary orbital motifs:

- * **The Lamb** - Sacrificial imagery (Genesis -> Revelation)
 - * **Wood** - Tree/Cross imagery (Eden -> Golgotha)
 - * **Silence** - Divine silence and still voice
 - * **The Binding** - Bondage and release
 - * **Water** - Creation through baptism
 - * **Fire** - Divine fire (Burning bush -> Pentecost)
 - * **Blood** - Blood covenant
 - * **Bread** - Manna to Eucharist
 - * **Shepherd** - Shepherd imagery
 - * **Stone** - Bethel to cornerstone
-

[BOOK] Hermeneutical Principles

From Hermeneutical.txt:

- * **Inevitable Judgment** - Constant background sense of approaching catastrophe
 - * **Emotional Honesty** - Preserve each event's native mood
 - * **Anti-Flattening** - Don't darken individual scenes
 - * **Pattern Pressure** - Let recognition do the work
 - * **Temporal Dislocation** - Memory as dread carrier
 - * **Load-Bearing Contrast** - Sharp contrasts at structural joints
 - * **Haunting over Foreshadowing** - Feel followed by what's been seen
-

■ Configuration

All settings in `config/settings.py`:

```
from config.settings import config

# Database
config.database.host
config.database.port

# Fourfold sense weights
config.fourfold_sense.literal_weight # 0.30
config.fourfold_sense.allegorical_weight # 0.25

# Thread density
config.thread_density.target_minimum # 18
config.thread_density.target_maximum # 22

# Orbital resonance
config.orbital_resonance.harmonic_ratios # [0.5, 0.833, 0.9375]
```

[TEST] Development

Documentation

For detailed implementation and workflow guidance, see:

- * docs/IMPLEMENTATION_GUIDE.md - Complete system documentation
- * docs/WORKFLOW_GUIDE.md - End-to-end processing workflow

Running Tests

```
pytest tests/
```

Code Style

```
# Format code
black scripts/ tools/ config/
# Lint
flake8 scripts/ tools/ config/
```

[NOTE] License

[Your License Here]

Acknowledgments

Based on the Orthodox Christian exegetical tradition, drawing from:

- * Patristic commentaries (Augustine, Chrysostom, Origen, etc.)
 - * Liturgical hermeneutics
 - * The Stratified Foundation System methodology
 - * MASTER_PLAN.md specifications
-

The blood-red sky comes from the whole arrangement, not from repainting each star.