

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m  
SDK Document: Z14m

The Digital-AV Software Development Kit (SDK provides the foundation for a fully working bible application), with no external dependencies. In fact, a fully operational application can be implemented with fewer than 1000 lines of code, as demonstrated by the golang sources in this SDK. As the SDK provides everything required, including data and index files. Some developers have discovered that they can be up and running in under an hour. You can easily jumpstart your development project by working with the provided golang sources, or go all in from scratch with the programming language of your choice.

The Digital-AV is entirely file based. There are zero dependencies and zero language bias (all programming languages can read files). File formats defined in this document use a consistent naming convention: the extent of each data file reveals the format and record width. The table to the right, defines the various extents of files that compose the SDK.

File extents identify the format [binary or text] and the record width. Extents of binary-formatted files begin with one of: {dx = for data; ix = for indices}. Extents that end in a numeric-digit are indicative that the files are fixed-width. The digit represents the count of 32-bit segments per record. As AV-Writ data files are available in three variants, and each variant's record width is clearly identified

by its file extent: .dx5, .dx4, and .dx1. Again, that digit reveals the count of 32-bit segments per record. For example, dx4 contains four 32-bit segments per record. This multiplies out to 128 bits, which equates to a fixed record width of 16 bytes. To be clear, dx4 does not mean every field is 32-bits; it is just a convenient shorthand for depicting the record width for SDK files containing fixed-width records. An "i" instead of a final numeric digit indicates that the binary file contains a built-in index, and that each record is variable width. Dedicated index file extents begin with .ix\_ and optionally provide index-assisted access to fixed length records. Index files are either 32-bytes [\*ix8], 8-bytes [\*ix2], or 4-bytes [\*ix1].

| File Extent | File Type    | Record Size |
|-------------|--------------|-------------|
| *.dx5       | data         | 20 bytes    |
| *.dx4       | data         | 16 bytes    |
| *.dx1       | data         | 4 bytes     |
| *.ix8       | index        | 32 bytes    |
| *.ix2       | index        | 8 bytes     |
| *.ix1       | index        | 4 bytes     |
| *.dxi       | data + index | variable    |
| *.bom       | text         | text        |
| *.ascii     | text         | text        |

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m  
SDK Document: Z14m

## Digital-AV – Detailed description of file layouts

The weightiest data files are those named AV-Writ.\*; these data files contain the stream of words for each verse of each chapter for each book. As these are not text files, the records are quite compact and some integer fields are lookups into other binary files. In essence, the entire set of binary files implement an efficient database of word embeddings, designed to be compactly manifested in RAM. The AV-Writ.\* files with the widest record-width are obviously also the most information rich.

AV-Writ.dx5 (4 × uint16 + 2 × uint16 + 2 × bytes + uint16 + uint32; 160 bits)

| Record<br>0 bits                             | Hebrew   Greek<br>(4 × 16 bits)                | Verse<br>16 bits | Caps<br>2 bits | WordKey<br>14 bits  | Punc<br>8 bits | Transition<br>8 bits | PN+WordClass<br>16 bits | POS<br>32 bits |
|--|--|------------------|----------------|---------------------|----------------|----------------------|-------------------------|----------------|
| 0  | 0x391C 0x0 0x0 0x0                             | 0x0000           | 0x8            | 0x0015 (in)         | 0x00           | 0xE8                 | 0x00E0                  | 0x40080470     |
| 1  | 0x391C 0x0 0x0 0x0                             | 0x0000           | 0x0            | 0x0136 (the)        | 0x00           | 0x00                 | 0x0D00                  | 0x00000094     |
| 2  | 0x391C 0x0 0x0 0x0                             | 0x0000           | 0x0            | 0x24F9 (beginning)  | 0x00           | 0x00                 | 0x4010                  | 0x000001DC     |
| ...  | << Beginning of Genesis 1 depicted above >>    |                  |                |                     |                |                      |                         |                |
| BDDB9  | 0x25A0 0x0 0x0 0x0                             | 0x77EA           | 0x8            | 0x0136 (the)        | 0x00           | 0xE0                 | 0x0D00                  | 0x00000094     |
| BDDBA  | 0x25A0 0x0 0x0 0x0                             | 0x77EA           | 0x8            | 0x2CB2 (revelation) | 0x00           | 0x00                 | 0x4010                  | 0x000001DC     |
| BDDBB  | 0x0978 0x0 0x0 0x0                             | 0x77EA           | 0x0            | 0x001D (of)         | 0x00           | 0x00                 | 0x0400                  | 0x80004206     |
| ...  | << Beginning of Revelation 1 depicted above >> |                  |                |                     |                |                      |                         |                |
| C0C91  | 0x1460 0x0 0x0 0x0                             | 0x797D           | 0x0            | 0x015C (you)        | 0x00           | 0x00                 | 0x20C0                  | 0x00083BBB     |
| C0C92  | 0x0F74 0x0 0x0 0x0                             | 0x797D           | 0x0            | 0x0036 (all)        | 0xE0           | 0x04                 | 0x0D00                  | 0x00000004     |
| C0C93  | 0x0119 0x0 0x0 0x0                             | 0x797D           | 0x8            | 0x018A (amen)       | 0xE0           | 0xFC                 | 0x8000                  | 0x8000550E     |
| << End of Revelation 22:21 depicted above >> |  |                  |                |                     |                |                      |                         |                |

AV-Writ.dx5 begins with **Greek & Hebrew** word keys, which correspond to Strong's numbers in the Old & New Testament. Each English word can have up to four Strong's numbers associated with it. Strong's numbers are an integer representation of the original Hebrew/Greek words from which the English words were originally translated (Refer to the Strong's Exhaustive Concordance for additional background information).

## Hebrew | Greek word representation

| Strong's #1                | Strong's #2                | Strong's #3                | Strong's #4                |
|----------------------------|----------------------------|----------------------------|----------------------------|
| 1 <sup>st</sup> Strong's # | 2 <sup>nd</sup> Strong's # | 3 <sup>rd</sup> Strong's # | 4 <sup>th</sup> Strong's # |

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m  
SDK Document: Z14m

It should be noted that while words in the Old Testament can have a maximum of four Strong's numbers representing the Hebrew associated with a single English word. The New Testament can only have a maximum of three Strong's numbers representing the Greek associated with a single English word. This is characteristic of the KJV translation, but four slots are reserved even for the greek to maintain a fixed record width across the entire bible.

**Verse**, is an inline index-pointer to the corresponding AV-Verse index.

The next sixteen bits can be thought of as two distinct fields: the first of those is **Caps**: these 2-bits identify whether to apply capitalization rules to the lexical word. 0x8\_\_\_ means to capitalize the first letter of the word (e.g. Lord). 0x4\_\_\_ means to capitalize all letters of the the word (e.g. LORD). Clearly, in English, the first letter of the first word of a sentence is capitalized, and these bits facilitate all such capitalization rules. When no bits are set, this indicated that the word should be represented exactly as it appears in the lexicon. The remaining 14-bits are called the **WordKey** (a lookup key for the AV-Lexicon). Incidentally, the lookup key is equally compatible with the Lexicons found in the older 2018 SDK.

## Punctuation Byte

| Description         | Bits |
|---------------------|------|
| PUNC::clause        | 0xE0 |
| PUNC::exclamatory   | 0x80 |
| PUNC::interrogative | 0xC0 |
| PUNC::declarative   | 0xE0 |
| PUNC::dash          | 0xA0 |
| PUNC::semicolon     | 0x20 |
| PUNC::comma         | 0x40 |
| PUNC::colon         | 0x60 |
| PUNC::possessive    | 0x10 |
| PUNC::closeParen    | 0x0C |
| MODE::parenthetical | 0x04 |
| MODE::italics       | 0x02 |
| MODE::Jesus         | 0x01 |

## Capitalization bits and WordKey

| Description                | Bit Pattern (Hex)                |
|----------------------------|----------------------------------|
| English Word               | 0x3FFF (mask for lexicon lookup) |
| 1 <sup>st</sup> Letter Cap | 0x8000 (example: Lord)           |
| All Letters                | 0x4000 (example: LORD)           |

The next field is the **Punctuation** byte. Each word can have certain punctuation applied either as a prefix to the word, or alternatively as a suffix. An example of prefix punctuation is an

open parenthesis. There are numerous examples of suffix punctuation, such as period, comma, or close parenthesis. The punctuation byte also has bits to represent italicized words in the text and even mark the words spoken by Jesus, which some bibles represent as red-colored text.

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m  
SDK Document: Z14m

## Person/Number (4 bits)

| Description            | Left-Most Nibble |
|------------------------|------------------|
| Person bits            | 0x3--- (0b--11)  |
| Number bits            | 0xC--- (0b11--)  |
| Indefinite             | 0x0--- (0b--00)  |
| 1 <sup>st</sup> Person | 0x1--- (0b--01)  |
| 2 <sup>nd</sup> Person | 0x2--- (0b--10)  |
| 3 <sup>rd</sup> Person | 0x3--- (0b--11)  |
| Singular               | 0x4--- (0b01--)  |
| Plural                 | 0x8--- (0b10--)  |
| WH*                    | 0xC--- (0b00--)  |

In AV-Writ.dx5 and in AV-Writ.dx4, Person/Number (PN) is the left-most nibble of the WordClass field. PN applies to pronouns and verb casing. Whereas Modern English is not morphologically rich when it comes to verb case, Early Modern English was slightly richer with additional pronouns and verb cases for Second-Person Singular and Third-Person Singular, each distinct from the Early Modern Plural counterparts. The Digital-AV captures and preserves all these distinct case markings. For instance, **thy** is second-person singular whereas Early Modern English **you** is always plural form of this pronoun. AV-SDK retains the markings for both person and number.

Transition bits are a composition of Verse-Transitions and Segment-Markers. These represent a compact mechanism for data file traversal, obviating the need for leveraging additional index files. The five left-most bits mark book, chapter, and verse

## Verse Transitions

| Description        | 5-bits |
|--------------------|--------|
| EndBit             | 0x10   |
| BeginningOfVerse   | 0x20   |
| EndOfVerse         | 0x30   |
| BeginningOfChapter | 0x60   |
| EndOfChapter       | 0x70   |
| BeginningOfBook    | 0xE0   |
| EndOfBook          | 0xF0   |
| BeginningOfBible   | 0xE8   |
| EndOfBible         | 0xF8   |

transitions. The three right-most bits mark linguistic segmentation [sentence and/or phrase] boundaries. In this edition of the SDK, these boundaries are interpreted based upon a combination of verse transitions and punctuation.

\* **his** is used ambiguously in the Authorized Version for third-person-singular pronouns. **his** is either masculine or neuter (**its** appears just once in the sacred text). Therefore, **his** can neither be uniformly marked as masculine, nor neuter. Instead, we mark the genitive pronoun **his** as non-feminine.

## WordClass (12 bits)

|                         |        |
|-------------------------|--------|
| NounOrPronoun           | 0x-03- |
| Noun                    | 0x-01- |
| Noun: unknown gender    | 0x-010 |
| Proper Noun             | 0x-03- |
| Pronoun                 | 0x-02- |
| Pronoun: Neuter         | 0x-021 |
| Pronoun: Masculine      | 0x-022 |
| Pronoun: Non-feminine*  | 0x-023 |
| Pronoun: Feminine       | 0x-024 |
| Pronoun/Noun: Genitive  | 0x-0-8 |
| Pronoun: Nominative     | 0x-06- |
| Pronoun: Objective      | 0x-0A- |
| Pronoun: Reflexive      | 0x-0E- |
| Pronoun: no case/gender | 0x-020 |
| Verb                    | 0x-1-- |
| to                      | 0x-200 |
| Preposition             | 0x-400 |
| Interjection            | 0x-800 |
| Adjective               | 0x-A00 |
| Numeric                 | 0x-B00 |
| Conjunction             | 0x-C0- |
| Determiner              | 0x-D0- |
| Particle                | 0x-E00 |
| Adverb                  | 0x-F00 |

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

There are two additional trimmed down versions of the AV-Writ files which contain subsets of AV-Writ.dx5. These can be used for more memory constrained implementations or utilized where the additional data fields are not needed.

## AV-Writ.dx4 (4 × uint16 + 2 × uint16 + 2 × bytes + uint16 = 128 bits)

| Record<br>0 bits                             | Hebrew   Greek<br>(4 × 16 bits)             | Verse<br>16 bits | Caps<br>2 bits | WordKey<br>14 bits | Punc<br>8 bits | Transition<br>8 bits | PN WordClass<br>16 bits |
|--|---|------------------|----------------|--------------------|----------------|----------------------|-------------------------|
| 0  | 0x391C 0x0 0x0 0x0                          | 0x0000           | 0x8            | 0x0015 (in)        | 0x00           | 0xEF                 | 0x00E0                  |
| 1  | 0x391C 0x0 0x0 0x0                          | 0x0000           | 0x0            | 0x0136 (the)       | 0x00           | 0x00                 | 0x0D00                  |
| 2  | 0x391C 0x0 0x0 0x0                          | 0x0000           | 0x0            | 0x24F9 (beginning) | 0x00           | 0x00                 | 0x4010                  |
| ...  | << Beginning of Genesis 1 depicted above >> |                  |                |                    |                |                      |                         |
| C0C91  | 0x1460 0x0 0x0 0x0                          | 0x797D           | 0x0            | 0x015C (you)       | 0x00           | 0x00                 | 0x20C0                  |
| C0C92  | 0x0F74 0x0 0x0 0x0                          | 0x797D           | 0x0            | 0x0036 (all)       | 0xE0           | 0x06                 | 0x0D00                  |
| C0C93  | 0x0119 0x0 0x0 0x0                          | 0x797D           | 0x8            | 0x018A (amen)      | 0xE0           | 0xFE                 | 0x8000                  |
| << End of Revelation 22:21 depicted above >> |   |                  |                |                    |                |                      |                         |

## AV-Writ.dx1 (uint16 + 2 × bytes = 32 bits)

| Record<br>0 bits                             | Caps<br>2 bits                              | WordKey<br>14 bits | Punc<br>8 bits | Transitions<br>8 bits |
|--|---|--------------------|----------------|-----------------------|
| 0  | 0x8   | 0x0015 (in)        | 0x00           | 0xEF                  |
| 1  | 0x0   | 0x0136 (the)       | 0x00           | 0x00                  |
| 2  | 0x0   | 0x24F9 (beginning) | 0x00           | 0x00                  |
| ...  | << Beginning of Genesis 1 depicted above >> |                    |                |                       |
| C0C91  | 0x0   | 0x015C (you)       | 0x00           | 0x00                  |
| C0C92  | 0x0   | 0x0036 (all)       | 0xE0           | 0x06                  |
| C0C93  | 0x8   | 0x018A (amen)      | 0xE0           | 0xFE                  |
| << End of Revelation 22:21 depicted above >> |   |                    |                |                       |

## Segment Markers

| Description    | 3-bits |
|----------------|--------|
| HardSegmentEnd | 0x04   |
| CoreSegmentEnd | 0x02   |
| SoftSegmentEnd | 0x01   |
| RealSegmentEnd | 0x06   |

*Hard Segments:* . ? !  
*Core Segments:* :  
*Real Segments:* . ? ! :  
*Soft Segments:* , ; ( ) --

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

The ascii variations of SDK files are intended to be informative, and are not considered to be core components of the inventory. Non-optimal performance would be achieved if a choice were made to develop against the ascii files instead of the binary formats. Yet, they are provided in the SDK to illuminate the formats of similarly organized binary files.

AV-Book.ix8 (2 × byte + UInt16 + 16 bytes + 12 bytes = 32 bytes)

| Record<br>0 bits | Book Number<br>byte | Chapter Count<br>byte | Chapter Index<br>UInt16 (2 bytes) | Book Name<br>16 bytes | Book Abbreviations<br>(comma separated; null padded)<br>12 bytes |
|------------------|---------------------|-----------------------|-----------------------------------|-----------------------|--|
| 0                | 1                   | 50                    | 0x000                             | Genesis               | Ge   |
| 1                | 2                   | 40                    | 0x032                             | Exodus                | Ex   |
| 2                | 3                   | 27                    | 0x05A                             | Leviticus             | Le   |
| ...              |                     |                       |                                   |                       |  |
| 65               | 66                  | 22                    | 0x4A4                             | Revelation            | Re   |

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

## AV-Chapter.ix2 (UInt32 + 2 × UInt16 = 64 bits)

| Record<br>0 bits         | Bible Index<br>32 bits | Verse Index<br>16 bits | Word Count<br>16 bits |
|--------------------------|------------------------|------------------------|-----------------------|
| 0x000<br>(genesis:1)     | 0x00000                | 0x0000                 | 0x31D                 |
| 0x001<br>(genesis:2)     | 0x0031D                | 0x001F                 | 0x278                 |
| 0x002<br>(genesis:3)     | 0x00595                | 0x0038                 | 0x2B7                 |
|                          | . . .                  |                        |                       |
| 0x4A2<br>(revelation:20) | 0xC058C                | 0x793F                 | 0x1DD                 |
| 0x4A3<br>(revelation:21) | 0xC0769                | 0x794E                 | 0x2ED                 |
| 0x4A4<br>(revelation:22) | 0xC0A56                | 0x7969                 | 0x23D                 |

## AV-Verse.ix1 (4 × byte = 32 bits)

| Record#<br>0 bytes | Book, Chapter, Verse, Words<br>4 bytes: BB:CC:VV:WordCnt |
|--------------------|--|
| 0x0000             | 1:1:1:10   |
| 0x0001             | 1:1:2:29   |
| 0x0002             | 1:1:3:11   |
|                    | ...  |
| 0x797B             | 66:22:19:44  |
| 0x797C             | 66:22:20:16  |
| 0x797D             | 66:22:21:12  |

In the beginning ...

And the Earth ...

And God said ...

And if any man ... are written in this book.

He which testifieth ... Even so, come, Lord Jesus.

The grace of our Lord ... be with you all. Amen

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m  
SDK Document: Z14m

The AV-Lemma file originally appeared in the 2017 Edition of the SDK. The original version obtained Lemmata from the NLTK Python library. Now Lemmata are obtained from the MorphAdorner Java server (MorphAdorner also performs all of the POS tagging). Incidentally, each Lemma ordinarily maps to multiple English words or lexemes, (e.g. The lemma ‘be’ corresponds to ‘are’, ‘were’, ‘is’, ‘art’, and ‘be’). Moreover, words like ‘run’ can function both as a verb and a noun.

## AV-Lemma.dxi (variable length)

| Part-of-Speech<br>32 bits | Word Key<br>16 bits | Word Class<br>16 bits | Count<br>16 bits | Lemma Array<br>UInt16[] (Word and/or OOV keys) |
|---------------------------|---------------------|-----------------------|------------------|--|
| 0x00000036                | 0x0001 (a)          | 0x0F00                | 1                | 0x0001   |
| 0x00000094                | 0x0001 (a)          | 0x0D00                | 1                | 0x0001   |
| 0x80004206                | 0x0001 (a)          | 0x0400                | 1                | 0x0001   |
| 0x01074F9C                | 0x0002 (i)          | 0x4080                | 1                | 0x0002   |
| ...                       |                     |                       |                  |  |
| 0x00003A1C                | 0x027A (elim)       | 0x4030                | 1                | 0x027A   |
| 0x000001DD                | 0x027B (elms)       | 0x8010                | 1                | 0x8304 (OOV: elm)                              |
| ...                       |                     |                       |                  |  |
| 0xFFFFFFFF                |                     |                       |                  |  |

Part-of-Speech needs to be considered when accessing the lemma utilizing AV-Lemma for looking up the lemma for a word. Lemmas contains a list of WordKeys and/or OOVKeys. When a Lemma is OOV<sup>1</sup>, it cannot be found in the Lexicon, but it can be found in the OOV table.

## AV-Lemma-OOV.dxi (lookup for OOV lemmas)

| OOV Key<br>16 bits | OOV Word<br>Length+1 bytes |
|--------------------|----------------------------|
| 0x8301             | aid\0                      |
| ...                |                            |
| 0x8F01             | covenantbreaker\0          |
| ...                |                            |

## OOV (composition by example)

| OOV<br>Marker<br>2 bits | OOV<br>Length<br>4 bits | OOV<br>Index<br>8 bits |
|-------------------------|-------------------------|------------------------|
| 0x8__                   | 0x_3__                  | 0x__01                 |

(binary of 0x8301 is b1000001100000001)

<sup>1</sup> OOV stands for “Out of Vocabulary”: Not all lemmas are in the AV-Lexicon; these OOV words can be looked up in the AV-Lemma-OOV table. As an example, “covenantbreakers” is in the KJV bible and therefore in the lexicon. However, covenantbreaker is not in the KJV bible (It is an example of an OOV word).



# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

AV-Lexicon provides the original AV orthographic representation and also the modern orthographic representation for each archaic lexeme of the AV text, along with a search-version of the lexeme that strips out all hyphens. What follows next is an array of one or more Part-of-Speech (POS) fields associated with the word. The POS array captures every POS combination encountered in the biblical text. Each POS entry is 5-bit encoded into a fixed 32-bit unsigned integer (Five-Bit encoding/decoding source is on gitlab)

## AV-Lexicon.dxi (data and index combined: variable length records)

| Rec#  | Entities<br>uint16 | Size<br>uint16 | 1<br>POS<br>UInt32 | 2<br>POS<br>UInt32   | 3<br>POS<br>UInt32 | ... | N<br>POS<br>UInt32 | Search<br>char []            | Display<br>char[]                   | Modern<br>char [] |  |
|-------|--------------------|----------------|--------------------|--|--------------------|-----|--------------------|------------------------------|-------------------------------------|-------------------|--|
| 1     | 0x0000             | N=4            | 0x00000094         | 0x00000036   | 0x0000000A         |     | 0x80004206         | a\0                          | \0                                  | \0                | Entities = { }<br>dt, av, j, pp-f      |
| 2     | 0x0000             | N=3            | 0x01074F9C         | 0x0000000A   | 0x01073F9C         |     |                    | i\0                          | \0                                  | \0                | Entities = { }<br>pns11, j, pno11      |
| 3     | 0x0000             | N=1            | 0x000002A8         |  |                    |     |                    | o\0                          | \0                                  | oh\0              | { }<br>oh                              |
| ...   |                    |                |                    |  |                    |     |                    |                              |                                     |                   |  |
| 366   | 0x8009             | N=2            | 0x00003A1C         | 0x000740FC   |                    |     |                    | adam\0                       | \0                                  | \0                | Entities =<br>{Man, City}<br>np1, npg1 |
| ...   |                    |                |                    |  |                    |     |                    |                              |                                     |                   |  |
| 1311  | 0x0000             | N=2            | 0x01073FBC         | 0x0000000A   |                    |     |                    | thou\0                       | \0                                  | you\0             | Entities = { }<br>pns21, j             |
| ...   |                    |                |                    |  |                    |     |                    |                              |                                     |                   |  |
| 12567 | 0x0000             | N=1            | 0x0000000A         |  |                    |     |                    | Mahershal<br>alhashbaz<br>\0 | Maher-<br>shalal-<br>hash-<br>baz\0 | \0                | Entities = { }<br>j                    |
| 12567 | 1416               | 12567          | 0xFFFFFFFF         | ← VersionNumber=Entities; Record-Count=Size; POS=end-of-file marker (0xFFFFFFFF) |                    |     |                    |                              |                                     |                   |  |

Entities = {Hitchcock=0x8000, men=0x1, women=0x2, tribes=0x4, cities=0x8, rivers=0x10, mountains=0x20, animals=0x40, gemstones=0x80, measurements=0x100}

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

## AV-WordClass.dxi (data and index combined: variable length records)

| WordClass<br>16 bits | Width<br>16 bits | 1 <sup>st</sup> POS<br>32 bits | 2 <sup>nd</sup> POS<br>32 bits | 3 <sup>rd</sup> POS<br>32 bits | ... | N <sup>th</sup> POS<br>32 bits |                         |
|----------------------|------------------|--------------------------------|--------------------------------|--------------------------------|-----|--------------------------------|-------------------------|
| 0x0010               | N=4              | 0x4000394E                     | 0x00003950                     | 0x40075AC7                     |     | 0x40075ACE                     | n-jn, njp, n-vvg, n-vvn |
| 0x00E0               | N=1              | 0x01074F9C                     |                                |                                |     |                                | p-acp                   |
| 0x0100               | N=29             | 0x00005842                     | 0x000B0893                     | 0x00005847                     |     | 0x00005ADA                     | Vbb, vbds, vbg, ... vvz |
| ...                  |                  |                                |                                |                                |     |                                |                         |

## AV-Names.dxi (data and index combined: variable length records)

| WordKey<br>16 bits            | 1 <sup>st</sup> Meaning | Delimiter | 2 <sup>nd</sup> Meaning | Delimiter | 3 <sup>rd</sup> Meaning | Delimiter | ... |
|-------------------------------|-------------------------|-----------|-------------------------|-----------|-------------------------|-----------|-----|
| AVLexicon WordKey for Aaron   | a teacher               |           | lofty                   |           | mountain of...          | \0        |     |
| AVLexicon WordKey for Abaddon | the destroyer           | \0        |                         |           |                         |           |     |
| AVLexicon WordKey for Abagtha | father of the...        | \0        |                         |           |                         |           |     |
| ...                           |                         |           |                         |           |                         |           |     |

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

## avx.go (golang source code)

avx.go implements a web-server (HTTP server) that provides the entire text of the AV bible utilizing AVX extensions, but still uses simple semantics. Version numbers for source code are respective of the SDK Document revision numbers. The first release of avx.go, which had been updated to the z-series SDK, was the Z081 golang source-code revision.

*Example of GoLang source in operation may be available at avbible.net:*

<https://avbible.net/avx/>

(the web-site above also utilizes NGINX as a reverse-proxy for HTTPS)

There are a couple of URLs used for testing and validation. They also illustrate how avx.go can be extended:

- <http://localhost:2121/>
- <http://localhost:2121/help>
- <http://localhost:2121/validate>

The / endpoint simply reports the release number of the optional avx.go web-server component. The /help endpoint provides primitive information about the web-service. /help can be easily replaced by developer. The /validate endpoint reports on the validity of data files in accordance with the bom (The “bom”, or bill of materials, is described in the section labelled AV-Inventory.bom later in this document). In addition to the administrative URL’s described above, here is a list of the foundational endpoints that provide the core functionality of avx.go:

### NOTES:

1. As the web-server is not hardened, it should be placed behind a reverse-proxy if exposed to the open Internet. This is a common pattern; Apache httpd, NGINX, Caddy, or IIS can easily be configured to serve as a reverse-proxy.
2. URL form #3 and #5 are discussed under the description of the \*.avspec format

1. <http://localhost:2121/avx/genesis>
2. <http://localhost:2121/avx/genesis/1>
3. <http://localhost:2121/avx/gen/1?sessionID>
4. <http://localhost:2121/avx/rev/22?sessionID=day&amen>
5. [http://localhost:2121/avx/rev/22?sessionID=\\$FFFFFFFFFFFFFF](http://localhost:2121/avx/rev/22?sessionID=$FFFFFFFFFFFFFF)
6. <http://localhost:2121/avx/css/sessionID.css>

All of these endpoints can be summarized as one of two types: getting the chapter of a book, or getting a CSS stylesheet. When no chapter is provided, chapter 1 is always implied. When no session identifier is provided, the resulting chapter request is decorated with the baseline stylesheet, named /css/AV-Stylesheet.css. When a session identifier is provided, the session number dictates the name of the CSS file that will decorate the chapter request. Moreover, avx.go can compile information into a CSS stylesheet. When a request is made for Genesis using the URL depicted in #3 above, a stylesheet becomes linked in the response to a stylesheet with the URL depicted in #8 above. A web-browser will make an immediate subsequent request to get the stylesheet. If /css/sessionID.css does not exist, avx.go will automatically compile a file named /css/sessionID.avspec. Similarly, but easier to understand in #4 above, the URL would generate CSS which would highlight the words **day** and **amen**. In order to maintain optimal performance, session identifiers are non-volatile. In order to overwrite a \*.css files and/or \*.avspec files, they must be manually deleted beforehand. Avx.go uses Z08 edition.

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14m

SDK Document: Z14m

## \*.avspec file format

| WordKey Count<br>UInt16 | Array of UInt16   |  |
|-------------------------|---|--|
| 0xn timer               | 0xn timer Count of WordKeys is followed by WordKey list [corresponds to AV-Lexicon] |  |
| BookChapter<br>UInt16   | Verse Count<br>byte (matching verses)   | Array of byte  |
| 0xbbcc                  | 0xkk  | 0xkk Count of matching verses is followed by an array of Verse numbers |
| ...                     |   |  |
| 0xbbcc                  | 0xjj  | 0xjj Count of matching verses is followed by an array of Verse numbers |
| 0x0000                  |   |  |

avx.go software ignores everything after the first record above. Only that first record defines the CSS file. And that first line is expanded word-for-word into highlights for each supplied wordkey. A slight variation here is that Strong's numbers will eventually also support highlighting. To highlight Strong's numbers, set the 0x8000 bit for Hebrew and the 0x4000 bit for Greek. The URL form that was depicted with this syntax, sessionID=\$FFFFFFFFFFFFFFF, is primarily intended for testing. Here, the hex digits that follow the dollar sign (\$) are expected to be expansions of the format described above (No record separators, just a representation of the raw bytes described above, in Big-Endian order).

## AV-Stylesheet.css (text file containing CSS for avx.go; optional)

This standard-format CSS stylesheet should be included when avx.go is utilized in your development. This optional stylesheet is included in the SDK, but it can be customized in any way by the web designer. However, the web designer should realize that any references in the CSS to image files will result in 404 errors unless support is explicitly added to avx.go by your development team. Finally, avx.go always links chapter output to the AV-Stylesheet.css stylesheet, even when a \*.avspec derived stylesheet is also specified.

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14<sub>m</sub>

SDK Document: Z14<sub>m</sub>

## AV-Inventory.bom (text file which identifies core inventory)

This is a text file that identifies the release and the delivered files for any given release. “bom” stands for bill of materials. For each artifact of the SDK, the bom lists each filename along with its corresponding MD5. The avx.go server implements a validation function that will read this file and report if the MD5’s agree with the SDK files on disk. This way corruption can be detected and as a revision can be validated in an automated fashion. Each Plate revision is released with a bom that should be deployed with the SDK. By doing this, streamlined plate revision checks are straightforward. MD5’s are calculated both on core SDK files, and on optional components, but not upon the bom itself. For each revision, it is recommended that you would copy AV-Inventory.bom to another location (e.g. AV-Inventory.Z08) along with the documentation associated with that release [this document may change over time].

### OVERALL PROJECT STATUS:

It’s an exciting time at AV Text Ministries, and if you want to lend a hand, let us know your technical skills and interests and we can help jumpstart you onto the team. Currently, AV Text Ministries is 100% volunteer, so if you don’t just have passion about the mission as your raw motivation, it might not be the best fit.

Finally, on the non-technical side of things, we would certainly welcome a ministry sponsor that would want to place AV Text Ministries under the banner of their own local church ministry. Check <http://avtext.org> to discover our overall vision.

### HOW THE DIGITAL-AV “PLATES” ARE AUTHORED:

Initially, various publicly available KJV texts were parsed and dutifully compared (comparing scripture with scripture [1 Corinthians 2:13]). That work produced the freeware program, AV-1995 for Windows; it was written in Delphi/Pascal and was maintained until the AV-2011. In 2008, the initial Digital-AV SDK was conceived and produced, harvesting much of the inner workings of AV-2008, utilizing RemObjects Oxygene/Pascal as a development platform and releasing it as open source. Later, AV-2011 was “compiled” using AV-2008 as a baseline. Subsequently, the 2017/2018 Editions were “compiled” using AV-2011 as a baseline. The Z07 revision of the SDK are baselined from AV-2018 edition using the K817 revision. C# is now the programming language of the SDK compiler; and the ancient pascal sources were finally retired (replaced by C# sources) in 2018. The Z-series SDK-compiler uses MorphAdorner<sup>2</sup> server, which is written in Java 1.6 a custom Python/Flask application wraps calls to NLTK<sup>3</sup>. Both servers are accessed via REST calls to local servers running on the program author’s network on localhost. POS tags are acquired from the MorphAdorner server and accordingly conform to the NUPOS tag set. Lemmatization is also acquired from the MorphAdorner server, with the NLTK Flask server utilized only if the MorphAdorner fails to return a lemma.

---

<sup>2</sup> <http://morphadorner.northwestern.edu/morphadorner/>

<sup>3</sup> <http://www.nltk.org>

# Digital AV SDK – Record layouts & File inventory

Z-series Edition / Revision: 1.4

## REVISION IDENTIFIERS

Digital-AV SDK: Z14<sub>m</sub>  
SDK Document: Z14<sub>m</sub>

### LICENSE REQUIREMENT:

- In order to comply with the MIT-style open-source license, please include AV-License.txt with your distribution of any file identified in this SDK. The text of that file as of 2020 is provided also at the bottom of this page.

*All SDK artifacts are on [github.com](https://github.com/kwonus/Digital-AV):*

<https://github.com/kwonus/Digital-AV>

### IMPROVEMENTS & CAVEATS:

- Fundamental SDK format has stabilized and is substantially similar to the 2017 and 2018 editions.
- Another field has also been added to AV-Writ which provides a precise Part-of-Speech representation for each word.
- The .dx? & .ix? extents now reflect the count of 32-bit segments for fixed-length records (earlier editions reflected the count of 16-bit segments).
- AV-Lemma has also been updated in the 2020 SDK. Moreover, the binary format of AV-Lexicon is also substantially different from earlier editions.
- Part-of-speech (POS) bits were introduced into the SDK with the HA29 release. As of the Z07 release, POS bits have been substantially revised as the SDK now uses MorphAdorner for part-of-speech marking instead of NLTK (NLTK doesn't recognize archaic verbs and pronouns, whereas MorphAdorner does).
- The sqlite lexicon has been eliminated from the SDK.

### ADDITIONAL RELEASE NOTES:

- The "Z-series" edition of the SDK introduced an updated revision number from earlier editions. Digital-AV revision numbers now use a three-digit character sequence, plus an optional suffix/subscript. All revision numbers now begin with the letter **Z**. The next two characters represent year and month of the revision. The character sequence is **Zym** where the first letter is always **Z**, indicating that this is the "Z-series" edition of the SDK (distinguishing it from older/legacy SDK editions); **y** represents the year, and **m** represents the month of the release. **y** encodes the year as a single base-36 digit; For example, (**y** = 0) represents 2020; (**y** = 1) represents 2021; (**y** = A) represents 2030; (**y** = K) represents 2040; (**y** = U) represents 2050. With respect to months, digits 1 through 9 are as expected; (**m** = A) is October; (**m** = B) is November; and (**m** = C) is December. An optional one-digit suffix/subscript may also be used. If the subscript is a Greek letter ( $\alpha$  or  $\beta$ ), then this identifies an alpha or beta release of the SDK. Otherwise, a suffix/subscript identifies the discrete date of the release, encoded in base-36; the 1<sup>st</sup> is 1, the 31<sup>st</sup> is q.
- Two revision numbers exist: Digital-AV SDK revision (aka, the "plate" revision) is the most significant set of files. Not that all files in this SDK are required to produce working bible software. Incidentally, the sample source code provided in avx.go implements a minimal set of SDK artifacts, while still providing access to the entire AV Bible text.
- Many of the binary files also have corresponding text files with an .ascii extent. These files are not provided for runtime execution. Instead, they should be considered as ancillary documentation to shed light, in painstaking detail, on the corresponding binary files.

### LICENSE:

Copyright (c) 1996-2021, Kevin Wonus

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice, namely AV-License.txt, shall be included with all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Additional information available at: <http://Digital-AV.org>, <http://AVText.org>, [info@avtext.org](mailto:info@avtext.org), [kevin@wonus.com](mailto:kevin@wonus.com)