Gary Griswold
gary@shortsands.com
513-508-6127
March 24, 2020
Revision 1

DBP v4 Data Dictionary

Descriptions of DBP data affected by Bible, Fileset load

This document describes the content of the DBP database for each table and column that is effected by the new load of the DBP. Additional tables will be added to this report as the logic to update those table nears deployment. Descriptions of the logic to update each column will be updated as the logic is refined through discussions.

bibles Table

All of the columns of the bibles table are returned by the /languages API query, while the /bibles API query returns bible_id, language_id, date, and copyright.

In the new load process, when retrieving LPTS data from the bibles table, I only retrieve records that contain text damid's which have a status of "active". This might be a controversial choice, and I am interested in clarifying whether there is a better process.

Most often, there is one LPTS record for each bible, but occasionally, there is more than one. For example, ENGKJV has stock numbers: C1KJV, C2KJV, N1KJV, N2KJV, and P2KJV. In those cases where there are multiple records, I lookup the field I am processing, such as Copyrightc in each of the records. If each record has the same value, then I use that value. If they do not have the same value, then in most cases, I display a warning to notify that a data administrator might wish to make a correction, and then I arbitrarily pick one value to update the database. In some cases, I might have logic to pick one value over another or combine the values in some meaningful way. Those details are discussed below with each field.

id - primary key

Bible id is often 6 characters, and often the first 3 characters match the ISO language code, and often the 3 following characters are an abbreviation of the version. It comes from the second part of a bible filename, e.g. text/bible_id/fileset_id. It also comes from LPTS where it is known as DBP_Equivalent, DBP_Equivalent2, and DBP_Equivalent3.

language_id

language_id is the primary key of the languages table. It is primarily found by doing a lookup in the languages_translations table using the LangName and ISO from the LPTS record. This query usually finds the correct record, but if the LangName is absent or does not match anything in the languages_translations table, then a second lookup is done in the languages table using only the ISO code. On a rare occasion the ISO code might also be missing. The current validation program reports on the absence of ISO and LangName, so this program does not produce any other error messages, but it could.

versification

The versification field contains information about the sequence of the books of the Bible in this specific Bible. It is returned by the /languages API query, but it seems unlikely that anyone would use it, because the bibles API query returns a much simpler, and more direct section called 'books', which lists the books of the Bible in their correct sequence, along with the names of the books in the language of the specific Bible, and a list of available chapters in each book.

The versification field is an example of a field where the DBP has field definitions that are entirely different than LPTS has for the same field. This DBP column currently contains the values: protestant, synodal, finnish, and messianic. Most records contain 'protestant', because the current load sets that value for all records. This column does not have a foreign key to any other table that would limit values, but it seems to be related to the books table which contains the following orders: protestant, luther, synodal, german, kjva, vulgate, lxx, orthodox, nrsva, catholic, finnish, messianic.

LPTS on the other hand, has fields OTOrder and NTOrder. The actual LPTS values for Old Testament are as follows: Masoretic-Christian, Masoretic-Tanakh, Panoramic, Septuagint, Traditional, Vulgate. The actual values from New Testament are as follows: Finnish, Plautdietsch, Russian, Traditional.

Since we do not have a way to change these OT, NT Orders from LPTS into the DBS versification field, there are a number of possibilities. 1) We could simply null this field on the assumption that it is not being used. 2) We could leave current contents, but not update this field. 3) We could change the content to contain both the OT and NT order values from LPTS and replace all current content with these LPTS values. 4) We could figure out how to translate the LPTS OT, NT values into the DBS values listed in the books table. At this moment, the logic of the new load will null out the versification field, but I open to anyone's ideas.

numeral_system_id

The numeral_system_id is the primary key of the numeral_systems table, which together with the numeral_system_glyphs table and the alphabet_numeral_systems table provides information of the numeric digits of the language of this Bible.

LPTS does not contain a data source for this field, and so the current load program sets each record with the value 'western-arabic'. Although, the table contains records with many other values that were originally set by DBS from some other data source.

The new program uses the script code (which follows below) in the bibles table to lookup the numeral_system_id in the alphabet_numeral_systems table by script code. There are two only cases where this does not return a single numeral_system_id. The 'Arab' script has three number_system_id's, but the dominant one is 'eastern-arabic'. And the script 'Deva' has two numeral_system_id's and the dominant one is 'devanagari'. For both these cases DBP should be updated with the dominant one. This is not entirely incorrect, because choosing one of the other number_system_id's for these languages should be based upon the country of residence of the User; it is not really an attribute of a specific Bible.

date

The date column is returned in the API by most requests that return any kind of Bible data. The current load program does not update this field in the bibles record. For those older records where it is populated, it almost always contains a 4 digit year that is the copyright year.

The process of the new program is to extract a 4 character date from the Copyrightc field. If the Copyrightc field is absent, then it gets the information from the Volumne_Name field.

scope

The scope field is only returned by the /languages API query as part of a bibles record. The contents of the current database does not follow any consistent pattern. The following is a list of current values: A, C, FB, FBA, Latn, NT, NTP, OT, OTA, P, PT. It seems like the best use of this field would be to contain the same set of allowed values that is in the bible_filesets.set_size_code. This is as follows: OT, OTP, NT, NTP, OTNTP, OTPNT, OTNT. What the new process does is to retrieve the set_size_codes for the text filesets of this bible_id, and use them here for scope. When there are different size code for different filesets that are part of one bible, then the logic does a logical union to produce a single value. For example, if we had OTP, NTP, and NT, the union would be OTPNT.

script

The script code is returned by the /languages and /bibles API queries. It is sourced from the the LPTS fields: _x0031_Orthography, _x0032_Orthography, _x0033_Orthography. A bible_id in an LPTS record is either DBP_Equivalent, DBP_Equivalent2, or DBP_Equivalent3. When a bible_id is found in a DBP_Equivalent, the associated index 1, 2, or 3 is remembered as associated with the bible_id. For script that index is used to retrieve orthography. For example, a DBP_Equivalent2 would get its script code from _x0032_Orthography. This is the value the new process will insert into the script field.

I have also written code that can determine the script code by looking at a bible's HTML or USX page, and finding the unicode values for a sample of characters in the file. When file upload capabilities are added to the load, I think this later process should replace getting script data from LPTS. Instead, it could produce a warning when the value found in the text and the value in LPTS are different.

derived

The derived column is returned by the /languages API query. The DBP database currently has this data for only 97 row. For all others, this column is null or blank. For those 97 rows, it appears to have excellent information, but there is no source in LPTS to update this information. The best solution for the new program is to ignore this field so that the current contents are not lost.

copyright

The copyright column is returned by the /languages and /bibles API query. It is the bible_fileset_copyrights table that contains the full copyright message that is included in the App. In the current system this column contains inconsistent data. Most records in the current system contain either: BY-NC-ND, or Public Domain, and 250 rows contain a more descriptive copyright message. Since the other columns, such a script, and numeral_system_id refer specifically to the text copy of the bible. I have chosen to use the Copyrightc LPTS field (text copyright) to populate this column.

priority

The priority column is only returned by the /languages query. There are only 45 rows where this field has a non-zero value. For the rows where the value is set, it appears to refer to a sequence that should order versions of a language. It is possible that the current BibleApp is using this for that purpose. This field has no source from LPTS. It appears these values were manually set. The new process will ignore this field, allowing it to continue to be manually set as needed.

reviewed

The reviewed column is returned by the /languages query. About 95% of this column have a value of 1 and 5% have a value of 0. I have no knowledge of the actual meaning of this column. There is no field in LPTS that I think could update this column. I have left it untouched by the new load. Or, it could be set to its default value of 0 if we expect to remove it from the database when that is convenient.

notes

The notes column is returned by the languages query. The current DBP database has only 4 rows where this column contains data. An example of its contents for bible_id ENGMNT is

"Greek text of Hermann von Soden". There is no known source for this field in LPTS. The new load will leave this column untouched, or it could null out the value if we expect to remove it from the database in the future.

bible_fileset_copyrights table

The bible_fileset_copyrights table only appears in the /bible/filesets/{fileset_id}/copyright API request. The copyright data in this table is displayed in the App. The logic used here to retrieve records from LPTS is a bit more permissive than what is described for the bibles table. This logic will retrieve LPTS records that have any damId status, whether it is live or something else.

For text filesets, there will sometimes be more than one 10 digit text damid, and sometimes more than one stock_number LPTS record. When this happens, the program looks for one that is live, and selects that one. Or, if there is more than one that is live, it will arbitrarily pick one. Or, if none are live, it will arbitrarily pick one of the records that has a damId that is not live, and use its copyright data. This logic does not currently generate warnings like the bibles table, when there are multiple records with different content.

copyright_date

The process searches for a 4 digit year in the copyright field, and uses it to update the copyright_date. If the fileset_id is text, then Copyrightc is the source. If the fileset_id is audio, the Copyrightp is the source. If the fileset_id is video then Copyright_Video is the source.

copyright

This field or copyright_description are is the copyright field that is used within the App. When the fileset type is text, Copyrightc is used. When the fileset type is audio, the contents are produced as "Text: {Copyrightc}\nAudio: {Copyrightp}". When the fileset type is video, the contents are produced as "Text: {Copyrightc}\nAudio: {Copyrightp}\nVideo: {Copyright_Video}"

copyright_description

This field contains the identical contents as copyright column. If we knew for certain which of these two fields was being used by the App, the other could be nulled.

open_access

This field is not being set, it always contains the default value of 1. The meaning of this field is not known.

bible_fileset_tags table

The contents of the bible_fileset_tags table does not appear anywhere in the API results that I have studied.

For text filesets, there will sometimes be more than one 10 digit text damid that matches the 6 character fileset_id, and sometimes more than one stock_number LPTS record. When this happens, the program looks for one that is live, and selects that one. Or, if there is more than one that is live, it will arbitrarily pick one. Or, if none are live, it will arbitrarily pick one of the records that has a damId that is not live, and use its copyright data. This logic does not currently generate warnings like the bibles table, when there are multiple records with different content.

hash_id

hash_id is the primary key of the bible_filesets table. It computed as a hash of the fileset_id, asset_id and set_type_code in the bible_filesets table.

name

The name field is a constant that defines the content of the description field. The possible values are bitrate, stock_no, and volume. The volume name is slated for deletion. There are also records with the name, timing_est_err, but these are set by the AudioHLS process, and are not affected by the DBP load.

language_id

This is set to a constant value 6414.

description

The contents of description varies depending upon the constant in the name field.

- bitrate - for audio filesets only, it contains 16kbps for filesets whose 10th-12th digits are 16, and otherwise contains 64kbps.
- stock_no - it contains the LPTS stock number of the record that data was extracted from.
- volume - it contains the LPTS Volume_Name. This field is slated for deletion.

admin_only

This is set to a constant value of null.

notes

This is set to a constant value of null.

iso

This is set to a constant value of 'eng'.