Feature Variations: New LookupVariation Mechanism

Skef Iterum, Adobe Inc.

March 11, 2024

This proposal adds an alternative to the FeatureVariationRecord mechanism in the Feature-Variations Table. The initial sketch of the new mechanism was developed by Behdad Esfahbod and Skef Iterum in discussion on GitHub.

The unnumbered "FeatureVariations Table" section at the start of the numbered "Feature variations" section is changed to the following (so that the last three of the current paragraphs are removed):

> A FeatureVariations Table describes how a given GSUB or GPOS Feature Table should be altered within specified regions of the design space of variable font, based on the value of *condition sets*, which are conjunctions of boolean *conditions*.
>
> The Feature List Table of a GSUB or GPOS table contains a list of offsets to Feature Tables which can be indexed either by tag or by offset in the list—the later being its "feature index". The FeatureVariations table provides two mechanisms for altering the lookups associated with a given feature index. One is the array of FeatureVariation-Records introduced in version 1.0 of the table. This system is feature-table oriented, providing an offset to a different feature table to be used when a set of conditions is met. The second mechanism is the array of LookupVariationRecords added in version 1.1. This system is lookup-oriented, allowing individual lookups to be chosen according to when a condition set is or is not met. A LookupVariationRecord provides a more compact encoding when more than a few lookups need to be changed.

/noindent The table definition becomes:

Type	Name	Description
uint16	majorVersion	set to 1
uint16	minorVersion	set to 1
uint32	featureVariationRecordCount	Number of feature variation records.
FeatureVariationRecord	featureVariationRecords[featureVariationRecordCount]	Array of feature variation records.
uint32	lookupVariationRecordCount	Number of lookup variation records. Added in version 1.1.
LookupVariationRecord	lookupVariationRecords[lookupVariationRecordCount]	Array of lookup variation records (sorted). Added in version 1.1.

The remaining material, not indented, is added to the end of the section.

LookupVariationRecord

Type	Name	Description
uint16	featureIndex	The feature table index to match (this is the sort key)
Offset32	featureLookupsTable	Offset to a FeatureLookupsTable

The LookupVariationRecord and its subtables provide an alternative mechanism for changing the lookups associated with a featureIndex. When the index for a feature is present in a LookupVariationRecord, the FeatureLookupsTable at the offset either supplements or replaces the default lookups associated with that index. The subtables of a FeatureLookupsTable are organized around individual lookups rather than whole tables, with individual or groups of lookups activated for the index when given condition sets apply or fail to apply.

(Every combination of lookups that can be expressed with a LookupVariationRecord can also be expressed with a FeatureVariationRecord, but the former will generally take up less space and be more straightforward to format and process than the latter when more than a few variations are specified.)

A typical FeatureVariations Table will contain either FeatureVariationRecords or LookupVariationRecords but not both. The primary reason for having both would be if LookupVariationRecords are used to determine the lookups that are applied for a given feature index but the featureParams (add cross reference) of one or more feature tables also need to change in certain regions of the font's design space. When a given feature index is listed in both a FeatureVariationRecord subtable and a LookupVariationRecord subtable, the featureParams are taken from the former and the set of lookups is taken from the latter.

Because LookupVariationRecords specify the set of lookups that apply at the default location just as it does with any other location, it is strongly recommended that the lookups active for the default location according to a FeatureLookupsTable match the list of lookups in the Feature Table of the feature index it corresponds to

FeatureLookupsTable

Type	Name	Description
uint16	majorVersion	set to 1
uint16	minorVersion	set to 0
uint16	flags	FeatureLookups qualifiers — see below
uint32	lookupConditionCount	Number of LookupCondition records.
LookupConditionRecord	lookupConditionRecord[conditionCount]	Array of LookupCondition records.

The FeatureLookupsTable provides offsets to a list of LookupConditionRecords that affect the associated featureIndex. Because all LookupConditionRecords are evaluated, they can be in any order.

Flags can be assigned to indicate certain uses or behaviors for a given FeatureLookups table. The following flags are defined.

Mask	Name	Description
0x0001	ADD_DEFAULT_LOOKUPS	When this bit is "on" the lookups in the default feature table for the index are added to the lookup set. Otherwise only the lookups specified by the LookupConditionRecords are included in the set.
0xFFFE	Reserved	Reserved for future use — set to 0.

LookupCondition Record

Type	Name	Description
Offset32	conditionSetOffset	Offset to a condition set table
Offset32	trueLookupIndexListOffset	Offset to a LookupIndexList table to add to the set when all conditions are true (0 if unused)
Offset32	falseLookupIndexListOffset	Offset to a LookupIndexList table to add to the set when at least one condition is false (0 if unused)

The LookupConditions table is equivalent to an if/else structure. When all conditions in the set are true all lookups from the trueLookupIndexList are added to the set of lookups corresponding to the feature index. When at least one is false the lookups from the falseLookupIndexList are added to that set. Either entry (but not both) can be disabled by setting it to 0. As with other condition sets, a 0 offset indicates the set is always true, and therefore the entries from the trueLookupIndexSet will be added.

LookupIndexList Table

Type	Name	Description
uint16	lookupIndexCount	Number of LookupList indices in this table.
uint16	lookupIndices[lookupIndexCount]	Array of indices into the lookup list.

This table simply encodes an array of lookupIndices to be added to the set.

Explanation of FeatureVariationsTable processing

The following is an example of how to process the FeatureVariationTable is included to clarify the relationship between the records and subtables. It is not prescriptive.

Recall that the "default" feature table corresponding to featureIndex (add cross reference) contains an offset to a `featureParams` table and a list of lookupList indices. Either or both of these can be substituted in relation to the chosen position in design space in the following way:

1. Process the featureVariationRecords:

   a. Evaluate the condition set of each FeatureVariationRecord in order until every condition of one set evaluates to true.
   b. If there is such a record, associate each feature index listed with its new Feature table offset

2. For each "active" feature index with a LookupVariationRecord:

   a. Allocate an empty feature table structure to use for that index.
   b. Copy any FeatureParams from the current feature table (either the replacement from step 1 if there is one, or the original Feature table for this index).
   c. If ADD_DEFAULT_LOOKUPS is set, copy list of lookups from current feature table into the set for this feature.
   d. For each LookupCondition record:

      i. If all conditions are true set o = trueLookupIndexSetOffset
      ii. Otherwise set o = falseLookupIndexSetOffset
      iii. Copy each lookup in the LookupIndexSet at o into the set for this feature

3. For each "active" feature index without a LookupVariationRecord:

   a. Use the current Feature table for that featureIndex (either the replacement from step 1 if there is one, or the original Feature table).