6 The Justification Table (JSTF)

The Justification table (JSTF) provides font developers with additional control over glyph substitution and positioning in justified text. Text-processing clients now have more options to expand or shrink word and glyph spacing so text fills the specified line length.

Overview

When justifying text, the text-processing client distributes the characters in each line to completely fill the specified line length. Whether removing space to fit more characters in the line or adding more space to spread the characters, justification can produce large gaps between words, cramped or extended glyph spacing, uneven line break patterns, and other jarring visual effects. For example:

> It's true that the basic standard of stuff out there is very poor, but that at which the press has been founded is a revelation of an ideological squeeze of every government attempt to allay the seemingly massed fears of downward trends spiralling in the local football game

Figure 6a. Poorly justified text

To offset these effects, text-processing clients have used justification algorithms that redistribute the space with a series of glyph spacing adjustments that progress from least to most obvious. Typically, the client will begin by expanding or compressing the space between words. If these changes aren't enough or look distracting, the client might hyphenate the word at the end of the line or adjust the space between glyphs in one or more lines.

To disguise spacing inconsistencies so they won't disrupt the flow of text for a reader, the font developer can use the JSTF table to enable or disable individual glyph substitution and positioning actions that apply to specific scripts, language systems, and glyphs in the font.

For instance, a ligature glyph can replace multiple glyphs, shortening the line of text with an unobtrusive, localized adjustment (see Figure 6b). Font-specific positioning changes can be applied to particular glyphs in a text line that combines two or more fonts. Other options include repositioning the individual glyphs in the line, expanding the space between specific pairs of glyphs, and decreasing the spacing within particular glyph sequences.

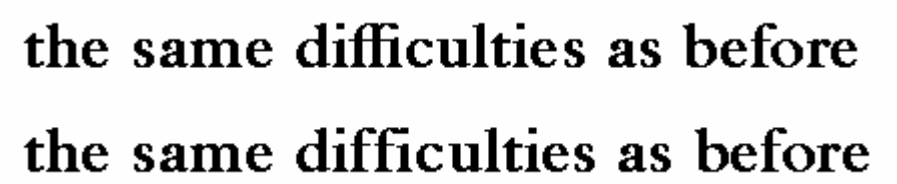

Figure 6b. JSTF shortens the top line of this example by using the "ffi" ligature

The font designer or developer defines JSTF data as prioritized suggestions. Each suggestion lists the particular actions that the client can use to adjust the line of text. Justification actions may apply to both vertical and horizonal text.

Table Organization

The JSTF table organizes data by script and language system, as do the GSUB and GPOS tables. The JSTF table begins with a header that lists scripts in an array of JstfScriptRecords (see Figure 6c). Each record contains a ScriptTag and an offset to a JstfScript table that contains script and language-specific data:

- A default justification language system table (DefJstfLangSys) defines script-specific data that applies to the entire script in the absence of any language-specific information.
- A justification language system table (JstfLangSys) stores the justification data for each language system.

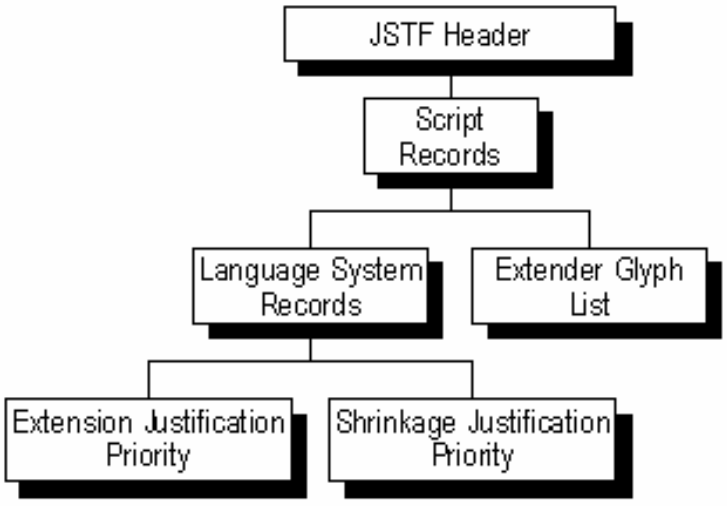

Figure 6c. High-level organization of JSTF table

A JstfLangSys table contains a list of justification suggestions. Each suggestion consists of a list of GSUB or GPOS LookupList indices to lookups that may be enabled or disabled to add or remove space in the line of text. In addition, each suggestion can include a set of dedicated justification lookups with maximum adjustment values to extend or shrink the amount of space.
The font developer prioritizes suggestions based on how they affect the appearance and function of the text line, and the client applies the suggestions in that order. Low-numbered (high-priority) suggestions correspond to “least bad” options.
Each script also may supply a list of extender glyphs, such as kashidas in Arabic. A client may use the extender glyphs in addition to the justification suggestions.
A client begins justifying a line of text only after implementing all selected GSUB and GPOS features for the string. Starting with the lowest-numbered suggestion, the client enables or disables the lookups specified in the JSTF table, reassembles the lookups in the LookupList order, and applies them to each glyph in the string one after another. If the line still is not the correct length, the client processes the next suggestion in ascending order of priority. This continues until the line length meets the justification requirements.

> **Note:** If any JSTF suggestion at any priority level modifies a GSUB or GPOS lookup that was previously applied to the glyph string, then the text processing client must apply the JSTF suggestion to an unmodified version of the glyph string.

The rest of this chapter describes the tables and records used by the JSTF table for scripts and language systems:

- Script information includes the JstfScript table (plus its associated JstfLangSysRecords) and the ExtenderGlyph table.
- Language system information includes the JstfLangSys table, JstfPriority table (and its associated JstfDataRecord), the JstfModList table, and the JstfMax table.

JSTF Header

The JSTF table begins with a header that contains a version number for the table (Version), a count of the number of scripts used in the font (JstfScriptCount), and an array of records (JstfScriptRecord). Each record contains a script tag (JstfScriptTag) and an offset to a JstfScript table (JstfScript).

> **Note:** The JstfScriptTags must correspond with the ScriptTags listed in the GSUB and GPOS tables.

Example 1 at the end of this chapter shows a JSTF Header table and JstfScriptRecord.

JSTF header

Type	Name	Description
fixed32	Version	Version of the JSTF table —initially set to 0x00010000

uint16	JstfScriptCount	Number of JstfScriptRecords in this table
struct	JstfScriptRecord [JstfScriptCount}	Array of JstfScriptRecords —in alphabetical order, by JstfScriptTag

JstfScriptRecord

Type	Name	Description
Tag	JstfScriptTag	4-byte JstfScript identification
Offset	⇨ JstfScript	Offset to JstfScript table —from beginning of JSTF Header

Justification Script Table

A Justification Script (JstfScript) table describes the justification information for a single script. It consists of an offset to a table that defines extender glyphs (ExtenderGlyph), an offset to a default justification table for the script (DefJstfLangSys), and a count of the language systems that define justification data (JstfLangSysCount).

If a script uses the same justification information for all language systems, the font developer defines only the DefJstfLangSys table and sets the JstfLangSysCount to zero (0). However, if any language system has unique justification suggestions, JstfLangSysCount will be a positive value, and the JstfScript table must include an array of records (JstfLangSysRecord), one for each language system. Each JstfLangSysRecord contains a language system tag (JstfLangSysTag) and an offset to a justification language system table (JstfLangSys). In the JstfLangSysRecord array, records are ordered alphabetically by JstfLangSysTag.

> **Note:** No JstfLangSysRecord is defined for the default script data; the data is stored in the DefJstfLangSys table instead.

Example 2 at the end of the chapter shows a JstfScript table for the Arabic script and a JstfLangSysRecord for the Farsi language system.

JstfScript table

Type	Name	Description
Offset	⇨ ExtenderGlyph	Offset to ExtenderGlyph table —from beginning of JstfScript table —may be NULL
Offset	⇨ DefJstfLangSys	Offset to Default JstfLangSys table —from beginning of JstfScript table —may be NULL
uint16	JstfLangSysCount	Number of JstfLangSysRecords in this table — may be zero (0)
struct	JstfLangSysRecord [JstfLangSysCount]	Array of JstfLangSysRecords —in alphabetical order, by JstfLangSysTag

JstfLangSysRecord

Type	Name	Description

Tag	JstfLangSysTag	4-byte JstfLangSys identifier
Offset	⇨ JstfLangSys	Offset to JstfLangSys table —from beginning of JstfScript table

Extender Glyph Table

The Extender Glyph table (ExtenderGlyph) lists indices of glyphs, such as Arabic kashidas, that a client may insert to extend the length of the line for justification. The table consists of a count of the extender glyphs for the script (GlyphCount) and an array of extender glyph indices (ExtenderGlyph), arranged in increasing numerical order. Example 2 at the end of this chapter shows an ExtenderGlyph table for Arabic kashida glyphs.

ExtenderGlyph table

Type	Name	Description
uint16	GlyphCount	Number of Extender Glyphs in this script
GlyphID	ExtenderGlyph [GlyphCount]	GlyphIDs —in increasing numerical order

Justification Language System Table

The Justification Language System (JstfLangSys) table contains an array of justification suggestions, ordered by priority. A text-processing client doing justification should begin with the suggestion that has a zero (0) priority, and then—as necessary—apply suggestions of increasing priority until the text is justified.

The font developer defines the number and the meaning of the priority levels. Each priority level stands alone; its suggestions are not added to the previous levels.

The JstfLangSys table consists of a count of the number of priority levels (JstfPriorityCnt) and an array of offsets to Justification Priority tables (JstfPriority), stored in priority order.

Example 2 at the end of the chapter shows how to define a JstfLangSys table.

JstfLangSys table

Type	Name	Description
uint16	JstfPriorityCnt	Number of JstfPriority tables
Offset	⇨ JstfPriority [JstfPriorityCnt]	Array of offsets to JstfPriority tables —from beginning of JstfLangSys table —in priority order

Justification Priority Table

A Justification Priority (JstfPriority) table defines justification suggestions for a single priority level. Each priority level specifies whether to enable or disable GSUB and GPOS lookups or apply text justification lookups to shrink and extend lines of text.

JstfPriority has offsets to four tables with line shrinkage data: two are JstfGSUBModList tables for enabling and disabling glyph substitution lookups, and two are JstfGPOSModList tables for enabling and disabling glyph positioning lookups. Offsets to JstfGSUBModList and JstfGPOSModList tables also are defined for line extension. Example 3 at the end of this chapter demonstrates two JstfPriority tables for two justification suggestions.

JstfPriority table

Type	Name	Description
Offset	⇨ ShrinkageEnableGSUB	Offset to Shrinkage Enable JstfGSUBModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ShrinkageDisableGSUB	Offset to Shrinkage Disable JstfGSUBModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ShrinkageEnableGPOS	Offset to Shrinkage Enable JstfGPOSModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ShrinkageDisableGPOS	Offset to Shrinkage Disable JstfGPOSModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ShrinkageJstfMax	Offset to Shrinkage JstfMax table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ExtensionEnableGSUB	Offset to Extension Enable JstfGSUBModList table —may be NULL
Offset	⇨ ExtensionDisableGSUB	Offset to Extension Disable JstfGSUBModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ExtensionEnableGPOS	Offset to Extension Enable JstfGSUBModList table —may be NULL
Offset	⇨ ExtensionDisableGPOS	Offset to Extension Disable JstfGSUBModList table —from beginning of JstfPriority table —may be NULL
Offset	⇨ ExtensionJstfMax	Offset to Extension JstfMax table —from beginning of JstfPriority table —may be NULL

Justification Modification List Tables

The Justification Modification List tables (JstfGSUBModList and JstfGPOSModList) contain lists of indices into the lookup lists of either the GSUB or GPOS tables. The client can enable or disable the lookups to justify text. For example, to increase line length, the client might disable a GSUB ligature substitution.

Each JstfModList table consists of a count of Lookups (LookupCount) and an array of lookup indices (LookupIndex).

To justify a line of text, a text-processing client enables or disables the specified lookups in a JstfModList table, reassembles the lookups in the LookupList order, and applies them to each glyph in the string one after another.

> **Note:** If any JSTF suggestion at any priority level modifies a GSUB or GPOS lookup previously applied to the glyph string, then the text-processing client must apply the JSTF suggestion to an unmodified version of the glyph string.

Example 3 at the end of this chapter shows JstfGSUBModList and JstfGPOSModList tables with data for shrinking and extending text line lengths.

JstfGSUBModList table

Type	Name	Description
uint16	LookupCount	Number of lookups for this modification
uint16	GSUBLookupIndex [LookupCount]	Array of LookupIndex identifiers in GSUB —in increasing numerical order

JstfGPOSModList table

Type	Name	Description
uint16	LookupCount	Number of lookups for this modification
uint16	GPOSLookupIndex [LookupCount]	Array of LookupIndex identifiers in GPOS —in increasing numerical order

Justification Maximum Table

A Justification Maximum table (JstfMax) consists of an array of offsets to justification lookups (Lookup) and a count of the defined lookups (Lookup). JstfMax lookups typically are located after the JstfMax table in the font definition.

JstfMax tables have the same format as lookup tables and subtables in the GPOS table, but the JstfMax lookups reside in the JSTF table and contain justification data only. The lookup data might specify a single adjustment value for positioning all glyphs in the script, or it might specify more elaborate adjustments, such as different values for different glyphs or special values for specific pairs of glyphs.

> **Note:** All GPOS lookup types except contextual positioning lookups may be defined in a JstfMax table.

JstfMax lookup values are defined in GPOS ValueRecords and may be specified for any advance or placement position, whether horizontal or vertical. These values define the maximum shrinkage or extension allowed per glyph. To justify text, a text-processing client may choose to adjust a glyph's positioning by any amount from zero (0) to the specified maximum.

Example 4 at the end of this chapter shows a JstfMax table. It defines a justification lookup to change the size of the word space glyph to extend line lengths.

JstfMax table

Type	Name	Description
uint16	LookupCount	Number of lookup Indices for this modification
Offset	⇨ Lookup [LookupCount]	Array of offsets to GPOS-type lookup tables —from beginning of JstfMax table —in design order

JSTF Table Examples

The rest of this chapter describes examples of all the JSTF table formats. All the examples reflect unique parameters described below, but the samples provide a useful reference for building tables specific to other situations.

The examples have three columns showing hex data, source, and comments.

Example 1: JSTF Header Table and JstfScriptRecord

Example 1 demonstrates how a script is defined in the JSTF Header with a JstfScriptRecord that identifies the script and references its JstfScript table.

Example 1

Hex Data	Source Comments
	JSTFHeader
	TheJSTFHeader ;JSTFHeader table definition
00010000	0x00010000 ;version
0001	1 ;JstfScriptCount
	;JstfScriptRecord[0]
74686169	"thai" ;JstfScriptTag
000C	ThaiScript ;offset to JstfScript table

Example 2: JstfScript Table, ExtenderGlyph Table, JstfLangSysRecord, and JstfLangSys Table

Example 2 shows a JstfScript table for the Arabic script and the tables it references. The DefJstfLangSys table defines justification data to apply to the script in the absence of language-specific information. In the example, the table lists two justification suggestions in priority order.

JstfScript also supplies language-specific justification data for the Farsi language. The JstfLangSysRecord identifies the language and references its JstfLangSys table. The FarsiJstfLangSys lists one suggestion for justifying Farsi text.

The ExtenderGlyph table in JstfScript lists the indices of all the extender glyphs used in the script.

Example 2

Hex Data	Source Comments
	JstfScript
	ArabicScript ;JstfScript table definition
000C	ArabicExtenders
	;ExtenderGlyph
0012	ArabicDefJstfLangSys;offset to DefJstfLangSys table
0001	1 ;JstfLangSysCount
	;JstfLangSysRecord[0]
50455220	"FAR " ;JstfLangSysTag
0018	FarsiJstfLangSys ;JstfLangSys
	ExtenderGlyph
	ArabicExtenders
	;ExtenderGlyph table definition
0002	2 ;GlyphCount
01D3	TatweelGlyphID
	;ExtenderGlyph[0]
01D4	LongTatweelGlyphID
	;ExtenderGlyph[1]
	JstfLangSys
	ArabicDefJstfLangSys;JstfLangSys table definition
0002	2 ;JstfPriorityCnt
000A	ArabicScriptJstfPriority1
	;offset to JstfPriority[0] table
001E	ArabicScriptJstfPriority2
	;offset to JstfPriority[1] table
	JstfLangSys
	FarsiJstfLangSys ;JstfLangSys table definition
0001	1 ;JstfPriorityCnt
002C	FarsiLangJstfPriority1
	;offset to JstfPriority[0] table

Example 3: JstfPriority Table, JstfGSUBModList Table, and JstfGPOSModList Table

Example 3 shows the JstfPriority and JstfModList table definitions for two justification suggestions defined in priority order. The first suggestion uses ligature substitution to shrink the lengths of text lines, and it extends line lengths by replacing ligatures with their individual glyph components. Other lookup actions are not recommended at this priority level and are set to NULL. The associated JstfModList tables enable and disable three substitution lookups.

The second suggestion enables glyph kerning to reduce line lenths and disables glyph kerning to extend line lengths. Each action uses three lookups. This suggestion also includes a JstfMax table to extend line lengths, called WordSpaceExpandMax, which is described in Example 4.

Example 3

Hex Data	Source Comments
	JstfPriority
	USEnglishFirstJstfPriority ;JstfPriority table definition
0028	EnableGSUBLookupsToShrink ;offset to ShrinkageEnableGSUB
	; JstfGSUBModList table
0000	NULL ;offset to ShrinkageDisableGSUB
	; JstfGSUBModList table
0000	NULL ;offset to ShrinkageEnableGPOS
	; JstfGPOSModList table
0000	NULL ;offset to ShrinkageDisableGPOS
	; JstfGPOSModList table
0000	NULL ;offset to Shrinkage JstfMax
	; table
0000	NULL ;offset to ExtensionEnableGSUB
	; JstfGSUBModList table
0038	DisableGSUBLookupsToExtend ;offset to ExtensionDisableGSUB
	; JstfGSUBModList table
0000	NULL ;offset to ExtensionEnableGPOS
	; JstfGPOSModList table
0000	NULL ;offset to ExtensionDisableGPOS
	; JstfGPOSModList table
0000	NULL ;offset to Extension JstfMax
	; table
	JstfPriority
	USEnglishSecondJstfPriority ;JstfPriority table definition
0000	NULL ;offset to ShrinkageEnableGSUB
	; JstfGSUBModList table
0000	NULL ;offset to ShrinkageDisableGSUB
	; JstfGSUBModList table

```
0000                                    NULL   ;offset to
                                        ShrinkageEnableGPOS
                                        ;  JstfGPOSModList table
001C                                    DisableGPOSLookupsToShrink
                                              ;offset to
                                        ShrinkageDisableGPOS
                                        ;  JstfGPOSModList table
0000                                    NULL   ;offset to Shrinkage
                                        JstfMax
                                        ;  table

0000                                    NULL   ;offset to
                                        ExtensionEnableGSUB
                                        ;  JstfGSUBModList table
0000                                    NULL   ;offset to
                                        ExtensionDisableGSUB
                                        ;  JstfGSUBModList table
002C                                    EnableGPOSLookupsToExtend
                                              ;offset to
                                        ExtensionEnableGPOS
                                        ;  JstfGPOSModList table
0000                                    NULL   ;offset to
                                        ExtensionDisableGPOS
                                        ;  JstfGPOSModList table
0000                                    NULL   ;offset to Extension
                                        JstfMax
                                        ;  table

                                        JstfGSUBModList
                                        EnableGSUBLookupsToShrink
                                              ;JstfGSUBModList table
                                        ;  definition
                                              ;  enable three ligature
                                        ;    substitution lookups
0003                                    3      ;LookupCount
002E                                    46     ;LookupIndex[0]
0035                                    53     ;LookupIndex[1]
0063                                    99     ;LookupIndex[2]

                                        JstfGPOSModList
                                        DisableGPOSLookupsToShrink
                                              ;JstfGPOSModList table
                                        ;  definition
                                              ;  disable three tight
                                        kerning
                                        ;    lookups
0003                                    3      ;LookupCount
006C                                    108    ;LookupIndex[0]
006E                                    110    ;LookupIndex[1]
0070                                    112    ;LookupIndex[2]

                                        JstfGSUBModList
                                        DisableGSUBLookupsToExtend
                                              ;JstfGSUBModList table
                                        ;  definition
                                              ;  disable three ligature
                                        ;    substitution lookups
0003                                    3      ;LookupCount
002E                                    46     ;LookupIndex[0]
0035                                    53     ;LookupIndex[1]
```

```
0063                                    99     ;LookupIndex[2]

                                        JstfGPOSModList
                                        EnableGPOSLookupsToExtend
                                               ;JstfGPOSModList table
                                        ;  definition
                                               ;  enable three tight
                                        kerning
                                        ;    lookups
0003                                    3      ;LookupCount
006C                                    108    ;LookupIndex[0]
006E                                    110    ;LookupIndex[1]
0070                                    112    ;LookupIndex[2]
```

Example 4: JstfMax Table

The JstfMax table in Example 4 defines a lookup to expand the advance width of the word space glyph and extend line lengths. The lookup definition is identical to the SinglePos lookup type in the GPOS table although it is enabled only when justifying text. The ValueRecord in the WordSpaceExpand lookup subtable specifies an XAdvance adjustment of 360 units, which is the maximum value the font developer recommends for acceptable text rendering. The text-processing client may implement the lookup using any value between zero and the maximum.

Example 4

Hex Data	Source Comments

```
                                        JstfMax
                                        WordSpaceExpandMax  ;JstfMax
                                        table definition
0001                                    1      ;LookupCount
0004                                    WordSpaceExpandLookup
                                               ;offset to Jstf Lookup[0]
                                        table

                                        Lookup
                                        WordSpaceExpandLookup      ;Jstf
                                        Lookup table definition
0001                                    1      ;LookupType
                                               ;  SinglePos Lookup
0000                                    0x0000 ;LookupFlag
0001                                    1      ;SubTableCount
0008                                    WordSpaceExpandSubtable
                                               ;offset to Subtable[0]
                                               ;  SinglePos subtable

                                        SinglePosFormat1
                                        WordSpaceExpandSubtable
                                               ;SinglePos subtable
                                        definition
0001                                    1      ;PosFormat
0008                                    WordSpaceCoverage   ;offset to
                                        Coverage table
0004                                    0x0004 ;ValueFormat
                                               ;  XAdvance only
0168                                    360    ;Value
```

```
                                        ;  XAdvance value
                                        ;  in Jstf, this is a max
                                    value
                                        ;    expand word space
                                    from zero
                                    ;     to this amount

                                    CoverageFormat1
                                    WordSpaceCoverage   ;Coverage
                                    table definition
0001                                1      ;CoverageFormat
0001                                1      ;GlyphCount
0022                                WordSpaceGlyphID
                                        ;GlyphArray[0]
```