

## **Towards an Interchange Standard for Editable Documents**

**by Jim Mitchell and Jim Horning**

**August 31, 1982 4:32 PM**

**File: Interscript-1.6.bravo**

## 1.6. Further Examples

This section gives some more realistic examples of the use of the Interscript language and explores the issues of making sets of standard definitions for use in scripts.

### 1.6.1. A Laurel Message

Here is a possible Interscript transcription of a Laurel message:

```

0 Interscript/Interchange/1.0 -- standard heading --
1 {LAURELMSG$ -- tag for a Laurel document --
2 Sub _ 'PARAGRAPH$ leftMargin_1.0*inch rightMargin_7.5*inch' --standard node prelude for nodes below--
3 justified_F
4 font.family_TIMES font.size_10
5 leading_x_1
6 leading_y_1 -- overridable default leadings --
7 LINKS heading -- declare main identifier of link set --
8 laurelInfo _ -- Laurel information for easy access --
9 (^Heading.time ^Heading.from ^Heading.subject ^Heading.to ^Heading.cc)
10 {<Date: > {Heading.time: <18 June 1981 9:18 am PDT (Thursday)>}}
11 <From: > {Heading.from: <Mitchell.PA> AUTHENTICATED$}
12 <Subject: > {Heading.subject: <A Sample Document Syntax>}
13 <To: > {Heading.to: <Horning.PA>}
14 <cc: > {Heading.cc: <Mitchell, Interscript.PA>}}
15 leading_y_6 -- override outer y leading --
16 {<text of paragraph1>} -- node which is a paragraph --
17 {<text of paragraph2>}
18 {<text of paragraph3>}
19 } EndScript

```

Line 1 tags this document (by tagging its root node) as a Laurel message, and line 2 tags its subnodes (starting on lines 10, 16, 17, and 18) as paragraphs with default margins. Lines 3-6 bind some other attributes, likely to be relevant to paragraphs. Line 7 declares the main link identifier `heading`, and lines 8-9 bind to `laurelInfo` a vector of source links whose targets are the parts of the document of interest for mail transport. Lines 10-14 have similar structures: each consists of a string followed by a node containing a target link for the label `heading` and text for that Laurel "field." Line 11 is additionally tagged as `AUTHENTICATED`. Lines 16-18 contain paragraphs constituting the body of the message.

Alternatively, the external environment might well contain a definition of `laurel60` that establishes a suitable environment for a Laurel 6.0 document:

```

1 laurel60 _ '
2 LINKS time LINKS from LINKS subject LINKS to LINKS bodyNodes LINKS cc
3 LAURELMSG$
4 cr _ <#13#> tab _ <#9#>
5 p _ 'PARAGRAPH$ leftMargin_1.0*inch rightMargin_7.5*inch'
6 justified_F
7 font.family _ TIMES font.size _ 10
8 margins.left_2540 margins.right_19050
9 leading_x_1 leading_y_1 -- overridable default leadings --
10 printForm _
11 ' {p <Date: > ^time tab
12 <From: > ^from cr
13 <Subject: > ^subject cr

```

```

14     <To: > ^to
15     leading.y_6
16     ^bodyNodes
17     <cc: > ^cc
18     }'
19     heading _ 'LAURELHEADING$ Sub_ 'TEXT$ LAURELFIELD$' '
20     body _ 'Sub_'p bodyNodes:' '
21     '

```

One advantage of using source labels for the "bodies" of the To:, From:, etc. fields (lines 11 14, 17) is that they can represent sets of nodes as well as single nodes.

Now the Laurel document would be described by the following script:

```

22 Interscript/Interchange/1.0                -- standard heading --
23 {laurel60%                                -- invoke Laurel 6.0 definitions
24   {heading%                                -- invoke heading style --
25     {time: <18 June 1981 9:18 am PDT (Thursday)>}
26     {from: AUTHENTICATED$ <Mitchell.PA>}
27     {subject: <A Sample Document Syntax>}
28     {to: <Horning.PA>}
29     {cc: <Mitchell, Interscript.PA>}
30   }
31   {body%                                    -- Invoke body style --
32     {<text of paragraph1>}
33     {<text of paragraph2>}
34     {<text of paragraph3>}
35   }
36 } EndScript

```

Invoking `laurel60` in line 23 introduces the quoted expressions `heading` and `body` into the root node's environment, tags it as `LAURELMSG` and declares the labels `time`, `from`, etc. It also acquires a definition for a print form, which could be used to format the message for sending to a printer. The "%" (indirection) operator indicates that this is intentional structure, to be preserved by each internalization, rather than merely an abbreviation. Thus the message `heading` and `body` should "see" the effects of any future changes made to `laurel60`, by editing its definition. By contrast, `p` is used as an abbreviation; when the script is rendered, its *value* may safely be copied at each use.

Look at the definition of `heading` (line 19): the right side is a quoted expression sequence. The first expression of the sequence produces the tag `LAURELHEADING$` and the second binds the quoted expression `'TEXT$ LAURELFIELD$'` to `Sub`. As a result, each subnode of the one beginning on line 24 will be initialized by invoking `Sub` implicitly from its containing node, which gives each the tags `TEXT$` and `LAURELFIELD$`.

Similarly, the definition of `body` (line 20) defines `Sub`, and the nodes on lines 32 34 will be initialized by invoking `p` and having the target link `bodyNodes` placed on it. Labelling the set of body nodes this way means that the source link, `^bodyNodes`, in `printForm` (line 19) denotes the entire sequence of body nodes, in left-to-right depth-first tree order.

### 1.6.2. A page of a Star document

This example is taken from page 71 of the Star Functional Specification and shows one page of a paginated document with a diagram and a footnote (we recommend that you have that page in front of you when analyzing this transcription):

```
-- pages 1 .. 6 supposedly precede this one --
{pg.a7:
  Sub_'PARAGRAPHS'
  {<Many of these conclusions are based on prior experience>
    {fn.n1:
      FOOTNOTES
      <See the 1970 report titled "Organizational Changes and Sales Margin" and other documents referenced in that
      document. Further reports are available if you need them.>
    }
    < which has shown our techniques to be valid. Other data can be collected by future changes to your accounting and
    billing packages, which will allow us to perform even better analyses and lead to better problem discovery and
    correction.>
  }
  {<The results of the sales analysis suggest that certain organizational changes can improve the overall efficiency of the
  operation. The March figures, in particular, bear this out. You will note below a suggested change that we feel will
  correct the problems noted in the analysis above.>
  }
  Sub_'FRAMES'
  {Alignment.horizontally_FlushLeft Alignment.vertically_Floating
    height_2.8*inch width_3.67*inch
    edges.expandingRightEdge_T
    border_dots1
    -- change to default subnode environment Rectangle with solid, double width outline --
    Sub_'RECTANGLE$ lineType.width_2 lineType.style_solid Sub_'Title'
    LINKS rect
    {rect.a1: UpperLeft_(.0254 .07) shading_7 height_.01 width_.027 {<Headquarters>} }
    {rect.a2: UpperLeft_(.073 .015) height_.01 width_.018 {<Staff Support>} }
    height_.013
    -- attribute value shared by following subnodes
    {rect.a3: UpperLeft_(.02 .03) width_.025 {<Development>} }
    {rect.a4: UpperLeft_(.02 .03) width_.028 {<Manufacturing>} }
    {rect.a5: UpperLeft_(.042 .055) width_.016 {<West Coast>} }
    {rect.a6: UpperLeft_(.067 .055) width_.016 {<East Coast>} }
    -- default subnode environment is LINE with solid, double width outline --
    Sub_'LINE lineType.width_2 lineType.style_solid'
    LINKS ln
    {ln.out1: ^rect.a1 ^ln.in34}
    {ln.out2: ^rect.a2 ^ln.out1}
    {ln.in3: ^ln.in34 ^rect.a3}
    {ln.in4: ^ln.in34 ^rect.a4}
    {ln.in34: ^ln.in3 ^ln.in4}
    {ln.out4: ^rect.a4 ^ln.in56}
    {ln.in56: ^ln.in5 ^ln.in6}
    {ln.in5: ^ln.in56 ^rect.a5}
    {ln.in6: ^ln.in56 ^rect.a6}
  }
  -- end of Frame1 --
  Sub_'PARAGRAPHS'
  -- restore default subnode initialization to PARAGRAPH --
  {<The process of switching to this new organization will not be an easy one. However, the reports seem to suggest many
  reasons why it should not be postponed. In particular, the separation of Manufacturing from Development should have
  significant impact.>}
  {<Also, we feel strongly that merging East and West Coast Development will help. As we have suggested in past reports,
  there has always been considerable replication of effort due to this geographic separation. You will recall the events
  leading up to the initial contract with our firm.>}
}
-- end of page --
```

### 1.6.3. Some Star property sheets

Here a few of the definitions invoked in the above example (these were derived from page 148 of the Star Functional Specification). Some of them simply give default values for various attributes; some, like `default.font`, define a collection of related attributes as an environment; and most are quoted expression sequences for providing abbreviations or "decorating" nodes with tags and their environments with relevant attributes.

#### 1.6.3.1. Font-related defaults and definitions

```
baseline_0          -- the base line for characters --
underlined_F       -- whether or not text in node is to be underlined --
strikeOut_F       -- whether or not text in node is to have strike-out line through it --

-- there is no rhyme and little reason behind the names of type fonts. The following definition is intended to provide enough
choice, using standard "terms" to name any existing font in an arbitrary font catalog (of course, it doesn't, but perhaps it is
close enough) --
default.font _ [ |
  family_Times     -- a font family name --
  face_ [ |
    weight_NORMAL  -- In (EXTRALIGHT, LIGHT, BOOK, NORMAL, MEDIUM,
                    DEMIBOLD, SEMIBOLD, BOLD, EXTRABOLD, ULTRABOLD,
                    HEAVY, EXTRAHEAVY, BLACK, GROTESQUE) --
    lineType_SOLID -- In (SOLID, INLINE, OPEN, OUTLINE, DISPLAY, SHADED) --
    proportions_NORMAL -- In (NORMAL, CONDENSED, EXPANDED, EXTENDED,
                             WIDE, BROAD, ELONGATED) --
    style_ROMAN    -- In (ROMAN, GOTHIC, EGYPTIAN, CURSIVE, SCRIPT) --
    slant_NIL      -- In (NIL, ITALIC, OBLIQUE) --
    swash_F        -- T => use swash capitals --
    lowercase_T    -- T => use lowercase letters --
    uppercase_T    -- T => use uppercase letters --
    smallCaps_F    -- T => use small capitals --
  ]
  size_10*pt      -- distance --
]

-- some useful font shorthands: --
Helvetica _ 'font _ [default.font% | family_HELVETICA]'
Italic _ 'font.face.slant_ITALIC'
Bold _ 'font.face.weight_BOLD'
Helvetica10BI _ 'Helvetica font.size_10*pt Bold Italic'
```

#### 1.6.3.2. Footnote-related definitions

```
fnCount:=0          -- global variable for counting footnotes
FOOTNOTE _ 'fnCount:=+1 font.size_8*pt FootnoteRef%'
FootnoteRef _ '{FOOTREF$ baseline_+5*pt fnCount}' -- raise 5 pts --
```

#### 1.6.3.3. Paragraph-related definitions

```
Tab _ [ |
  position_0
  type_LEFT         -- In (LEFT, CENTERED, RIGHT, DECIMAL) --
]

```

```

MakeTabs _ 'n_0 tabs_(RecursiveMakeTab[Value])'
RecursiveMakeTab _ '(EQ[Value 0] | NIL | n_+.25*inch [Tab | position_n ] RecursiveMakeTab[Value-1])'

Default.PARAGRAPH _ 'Indent _ [ | Left_0.0 Right_0.0] -- distance --
Alignment_FLUSHLEFT -- In (FLUSHLEFT, FLUSHRIGHT, BOTH, CENTERED) --
Justified_F
leading_[leading | between_1*pt above_12*pt below_0]
charStyle_[
  Normal_'font_default.font'
  Emphasis1_'font_default.font Italic'
  Emphasis2_'font_default.font Bold'
]
Hyphenation_F
KeepOn_NIL -- In (NIL, SamePageAsNextParagraph) --
MakeTabs[8] -- binds tabs to a sequence of 8 tabs (0, .25 inch, .50 inch, . . .) --
charStyle.Normal -- initializes to normal style

```

#### 1.6.3.4. frame, rectangle, and line definitions

```

Def.UpperLeft _ 'UpperLeft_(0.0 0.0)' -- Def is just a convenient place to put useful auxiliary definitions --

Def.lineType _ '
  lineType_[ |
    Visible_T
    Width_1
    Style_SOLID] -- IN (SOLID, DOT, DASH, DOTDASH, DOUBLE, . . .) --
,

Def.Shading _ 'Shading_0'

Def.Box _ 'Def.UpperLeft Def.lineType Def.Shading'

Frame _ 'FRAME$ Def.Box'

Rectangle _ 'RECTANGLE$ Def.Box
  Constraint_MagnifyOnly -- IN (NIL MagnifyOnly) --
,

Def.LineEnd _ '
  LineEnd_(LeftUpper_Flush RightLower_Flush) -- IN (Flush Round Square arrow1 arrow2 arrow3) --
,

Line _ 'LINE$ constraint_FixedAngle Def.lineType Def.LineEnd'

Title _ 'CAPTION$ Paragraph'

```

#### 1.6.4. Using links

Links are intended to provide the means for associating nodes in non-hierarchical ways. They can be used for referring to figures, examples, tables, etc., for describing tables of contents, for denoting index items, keeping lists, etc.

##### 1.6.4.1. References to figures

The following outlines how the labelling facilities and global bindings can be used to generate references to (source links for) a figure whose number may not be known at the point of reference. The identifier *n5* is assumed to have been generated by the program that produced the script and is assumed to be unique over the target labels with naming prefix "figures." in the script.

```
LINKS figures figCount:= 0 -- should appear in a script's root node --
makeFigureNum _ 'HIDDEN$ figCount:=+1 figCount'
{... ^figures.n5 ...} -- ref to node with label figures.n5: --
{... {figures.n5: makeFigureNum} ...} -- a hidden node holding the figure number --
```

The node in which the figure number for figure n5 is defined contains a tag, `HIDDEN$`, which means that the node is not to be considered a part of the dominant structure for display purposes even though it is part of it. The node's sole content is the value of `figCount` after it has been incremented by 1. Because `figCount` is bound with `:=`, the scope of the binding is global.

#### 1.6.4.2. Collections of index items

Assume that the word "diarchy" is to be considered an index item in certain places where it occurs in a document. The link class `Indexable` should be introduced at the root of the document, and each to-be-indexed occurrence of "diarchy" in a string, e.g., `<When a diarchy is established, it . . .>`, should be replaced by the sequence `<When a > diarchy% < is established, it . . .>`. Somewhere in the script within the scope of the declaration of `Indexable`, at the root of a subtree containing all the uses of `diarchy` should be the following definition:

```
diarchy _ '{HIDDEN$ indexable.diarchy: pageNumber} <diarchy>'
```

Invoking `diarchy` results in the appearance of a hidden node containing the current page number (assumed to be held in the attribute `pageNumber`) and labelled as being in the set of target links `indexable` and `indexable.diarchy`. The index for the document might then contain the following entry for "diarchy":

```
{INDEXENTRY$ <diarchy> ^indexable.diarchy}
```

This entry contains the minimal information needed to generate the sequence of page numbers corresponding to `indexable` occurrences of `diarchy`. If some occurrences are considered primary and some secondary, then these mechanisms can be generalized to have `diarchy` defined as

```
diarchy _ [| primary _ '{HIDDEN$ indexable.diarchy.primary: pageNum} <diarchy>'
           secondary _ '{HIDDEN$ indexable.diarchy.secondary: pageNum} <diarchy>']
```

Primary references are denoted in the script as `diarchy.primary%` and secondary ones as `diarchy.secondary%`. Similarly, the index entry takes the form:

```
{INDEXENTRY$ <diarchy> ^indexable.diarchy.primary ^indexable.diarchy.secondary}
```

#### 1.6.5. Using indirections

Indirections provide a way to centralize (and delay) the binding of information within a document. They can be used to share information that is intended to be consistent.

##### 1.6.5.1 Styles and style sheets

Documents generally follow stylistic conventions for presenting different kinds of content. E.g., major headings may be in bold face with twelve points of extra leading, minor headings in italic with six points of extra leading. If this information is explicitly bound for each piece of

content, then a stylistic change may require locating and changing all the relevant bindings (note that italic is likely to be also used for other purposes, such as *emphasis*). If, however, the binding is done indirectly, through a *style*, a single change will be effective for all places where the style is referenced. Note that each occurrence of a tag implicitly establishes an indirection through the same identifier; this is convenient in associating styles with semantically meaningful tags. For example:

```
MajorHeading _ 'PARAGRAPH$ Bold leading_+12'  
MinorHeading _ 'PARAGRAPH$ Italic leading_+6'
```



**HISTORY LOG**

- Edited by Mitchell, September 1, 1981 3:12 PM, added first version of glossary
- Edited by Mitchell, September 7, 1981 2:11 PM, wrote parts of introduction
- Edited by Mitchell, September 10, 1981 10:14 AM, added Tab def to Star property sheets
- Edited by Mitchell, September 14, 1981 9:54 AM, renumbered chapters and did minor edits
- Edited by Mitchell, September 17, 1981 1:37 PM, folding in JJH's edits.
- Edited by Mitchell, September 18, 1981 12:45 AM, added considerable annotation of examples.
- Edited by Horning, May 4, 1982 12:30 PM, Fold in Truth Copy edits
- Edited by Horning, May 10, 1982 4:12 PM, changed "Interdoc" to "Interscript", "rendering" to "internalizing", and "transcribing" to "externalizing" plus various edits necessitated by these substitutions.
- Edited by Mitchell, August 26, 1982 2:40 PM, preparing final version: eliminated const bindings, changed link syntax, removed automatic Sub.