User Externals for XML

Last update 03/09/2002

Changes to the Pagedef related Commands

PPFA limitations:

- XLAYOUTs with relative-x positioning cannot have:
  1. FIELD commands with x-positioning relative to the XLAYOUT (LPOS).
  2. FIELD ATTR (attribute) with x-positioning relative to the XLAYOUT (LPOS).
  3. FIELD commands with barcodes.
  4. DRAWGRAPHIC commands
  5. ENDGRAPHIC commands
  6. OBJECT subcommands
  7. SEGMENT subcommands
  8. OVERLAY subcommands
- Relative-x positioning on an XLAYOUT means relative to the current position. That is wherever the previous placed XLAYOUT left the current position after applying the XLAYOUT positioning, its fields and attributes. Graphics, barcodes, objects, segments, and overlays do not change the current position after they are originally positioned. For example, if you position a graphics box with a DRAWGRAPHIC command, the beginning of the box becomes the new current position but any movement in drawing the box does not change the current position. In another example, if you position a text field and then place the text, the end of the text becomes the new current position.
- Attributes are special FIELDs. The attribute is identified by name and the data printed is from the attribute value, not from the element content.
- If a FIELD is used for presenting any piece of data on an XLAYOUT, FIELDs must be used for all pieces of data to be presented on the XLAYOUT. Since an attribute is a special field, if you want to print both an attribute value and the element data you need to code the attribute field for the attribute value and a regular field for the element data.
- When SAME is coded as the XLAYOUTs' x-position there is a further deviation from the way it works for PRINTLINE and LAYOUT. SAME-x should mean same x-position as the previously coded XLAYOUT, but because of PSF and printer limitations, we can only compute this position if the previously coded XLAYOUT has absolute-x positioning. This limitation will only be in place while the PSF and printer limitations exist. So when SAME (x-pos) is coded and the previous XLAYOUT has relative-x-positioning, PPFA will change it to RELATIVE 0 and issue a warning message. A page definition will still be built.

PPFA XML notes:

Notes:

1. Leading and trailing blanks in the data are suppressed. Further, multiple imbeded blanks will be reduced to one blank.

Changes to the DEFINE command

The DEFINE command has a new subcommand "QTAG". This subcommand defines an XML data Qualifying tag. This makes the coding of an XLAYOUT command easier. One can give a local name to an XML data item and refer to that internal name on the XLAYOUT instead of the fully qualified name.

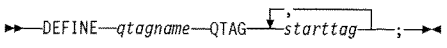

Figure 1. Syntax drawing for new DEFINE subcommand QTAG

QTAG Defines a local identifier for a qualified tag which can be used later in the page definition on a XLAYOUT command. A QTAG is a sequence of one or more start-tag names which taken together identify an XML data element. This is the logical equivalent of the "record IDentifier" on the LAYOUT command for a record formatting page definition. But, instead of identifying an entire record as the LAYOUT command does, the QTAG identifies a single XML data element.

When used, the local identifier makes the coding of an XLAYOUT command easier by allowing the use of a locally defined name instead of the fully qualified set of "start tags". It also makes the XLAYOUT command syntax similar to the LAYOUT command.

qtagname The internal name assigned to the fully qualified QTAG. This name can be used on the XLAYOUT command to identify the XML data item. This name is not case sensitive. It can be up to 16 characters long.

starttag An XML element name. This name must match exactly to the element name in the XML data. To preserve the case for the name, put it in quotes. Otherwise the name will be folded to upper case. If necessary, the name will be translated to the datatype specified or defaulted by the UDTYPE subcommand on the PAGEDEF command. For example, if the page definition is coded on an EBCDIC platform but the UTDTYPE specifies UTF8, PPFA will convert the start tags from EBCDIC code page 500 to UTF-8.

See the XLAYOUT command for an example of using a defined QTAG with an XLAYOUT command.

Changes to the Page Definition command

The PAGEDEF command has a new subcommand "UDType". This subcommand identifies the encodation of the users data, for example UTF-8. This will cause all qualified tags, attribute names, and FIELD and CONDITION text to be translated by PPFA if it does not match the platform data type. For example, on an OS/390 platform the input data type encodation is assumed to be EBCDIC code page 500. Any UDType that is not EBCDIC will cause PPFA to translate the above items.

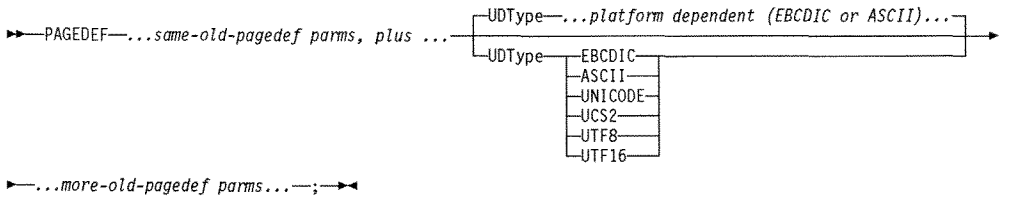

Figure 2. Syntax drawing for new PAGEDEF subcommand UDType

UDType

This subcommand identifies the encodation of the users data. If it does not match the platform data type, it will cause all qualified tags, attribute names, and FIELD and CONDITION text to be translated by PPFA to that data type. For example, on an OS/390 platform the PPFA page definition encoding is assumed to be EBCDIC code page 500. So that, if one specifies that their data type (UDType) is UTF8, PPFA will translate all field and condition text, start tags and attribute names from EBCDIC code page 500 to UTF-8.

The UDType is also passed on to the printer (or other presentation device) so that it can translate the data to the font encodation if that transformation is supported. Allowable combinations are explained in the documentation for the FONT command earlier in this chapter.

If UDType is not coded on the pagedef, it will default to either EBCDIC or ASCII according to the platform and no translation will be done.

EBCDIC Single byte EBCDIC code page 500.

ASCII Single byte ASCII code page 819.

UNICODE (or UCS2) Fixed two byte unicode without surrogates. This is also known as UCS-2.

UTF8 Unicode encoding form UTF-8.

UTF16 Unicode encoding form UTF-16. Note that the pagedef will be created in UTF-16BE (Big Endian). If the data is in UTF-16LE, PSF will translate it to UTF-16BE before processing.

New Command XLAYOUT

The XLAYOUT command is analogous in function to the PRINTLINE and LAYOUT commands. It specifies the printing of one XML data item. The item is identified on the XLAYOUT command by defined or explicit qualified tag. Think of this as the XLAYOUT name equivalent to the record-id on a LAYOUT command.

As PSF scans the data and finds an XML data item, it looks in the page definition for a match using the fully qualified tag. XLAYOUT differs from the LAYOUT command in essentially three ways plus the noted restrictions:

1. The name. In place of the LAYOUT name (DEFAULT or 'recid') use either a defined qualified tag (qtagname) or an explicit qualified tag (QTAG starttag,starttag,...)
2. An XLAYOUT cannot have a variable resource name.

3. The POSITION parameter. The position parameter will allow a RELATIVE or ABSOLUTE x-position. ABSOLUTE is the default.

   Note: The relative-x position function will have the following restrictions:

   a. Relative-x on an XLAYOUT will only be relative to the current x-position. That is wherever the previous placed XLAYOUT left the current position after applying the XLAYOUT positioning, its fields and attributes. Graphics, barcodes, objects, segments, and overlays do not change the current position after they are originally positioned. For example, if you position a graphics box with a DRAWGRAPHIC command, the beginning of the box becomes the new current position but any movement in drawing the box does not change the current position. This is different from text placement with attribute or text fields. For example, if you position a text field and then place the text, the end of the text becomes the new current position.
   b. If a FIELD is used for presenting any piece of data on an XLAYOUT, FIELDs must be used for all pieces of data to be presented by the XLAYOUT. For example: An attribute is a special field. So if you want to print both an attribute value and the element data, you will need to code both the attribute field for the attribute value and a regular field for the element data.
   c. XLAYOUTs with relative-x positioning cannot have:
      1) FIELD commands with x-positioning relative to the XLAYOUT (LPOS).
      2) FIELD ATTR (attribute) with x-positioning relative to the XLAYOUT (LPOS).
      3) FIELD commands with barcodes.
      4) OBJECT subcommands
      5) SEGMENT subcommands
      6) OVERLAY subcommands
      7) DRAWGRAPHIC commands
      8) ENDGRAPHIC commands

4. When SAME is coded as the XLAYOUTs' x-position there is a further deviation from the way it works for PRINTLINE and LAYOUT. SAME-x should mean same x-position as the previously coded XLAYOUT, but because of PSF and printer limitations, we can only compute this position if the previously coded XLAYOUT has absolute-x positioning. This limitation will only be in place while the PSF and printer limitations exist. So when SAME (x-pos) is coded and the previous XLAYOUT has relative-x-positioning, PPFA will change it to RELATIVE 0 and issue a warning message. A page definition will still be built.

All other XLAYOUT parameters are the same as the LAYOUT command parameters.

The following new syntax will be valid on the XLAYOUT command:

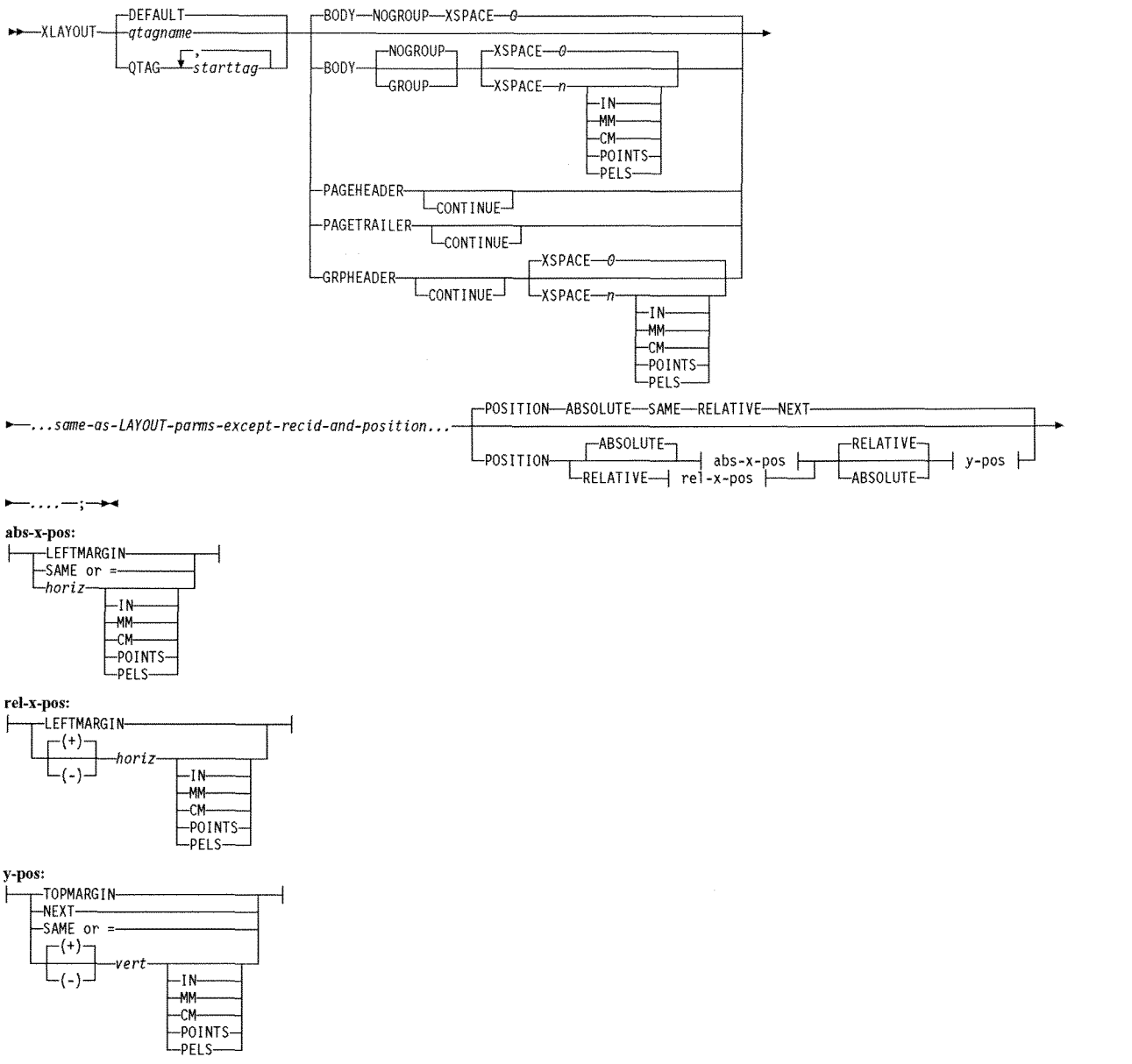

Figure 3. Syntax drawing for new XLAYOUT command

XLAYOUT Name This is a defined or explicit Qualified Tag

DEFAULT This keyword is used only when the XLAYOUT type is PAGEHEADER or PAGETRAILER, and no name is needed. Only one default PageHeader or PageTrailer can be specified in a Pageformat.

qtagname This is a defined Qualified Tag. It is defined by the "DEFINE qtagname QTAG" command at the beginning of the page definition.

QTAG starttag This is an explicit Qualified Tag. It is defined by coding a series of start tags separated by commas. A start tag is an XML data element name. Put the starttag in quotes if you want to preserve its case. Otherwise it will be folded to upper case.

Example of XML data with the associated Page Definition showing the use of both a defined and explicit QTAG:

```
<person>
  <name>
    <first>Johnny</first>
    <last>Johnson</last>
  </name>
</person>
```

```
PAGEDEF xxxx...;
  DEFINE lname QTAG 'person','name', 'last';
  ...
  Pageformat x ...
    XLAYOUT lname POSITION...
    ...
  Pageformat y ...
    XLAYOUT QTAG 'person','name','last' POSITION ...
  ...
```

In the above example, "person", "name", "last", and "first" are start tags, and the qualifying tag for the data item "Johnson" is "'person','name','last'". In the partial page definition, both of the XLAYOUT commands address the same XML data item, "Johnson".

BODY -- same same --

PAGEHEADER -- same same --

CONTINUE Continuation parameter. This indicates that this XLAYOUT is a continuation of the Page Header, Group Header, or Page Trailer definition. The formation of of the Page Header, Group Header, or Page Trailer may require the data from more than one data element. This is accomplished by specifying this continuation parameter.

PAGETRAILER -- same same --

GRPHEADER -- same same --

POSITION same as before except that RELATIVE can occur before the x-position.

Notes:

1. LEFTMARGIN and TOPMARGIN are absolute positions and override the keyword RELATIVE if coded.
2. NEXT is a relative position and overrides the y-position ABSOLUTE if coded.
3. Do not use the "+" or "-" for absolute positions.

horiz Specifies a number of units (inches, cm, mm, etc) used to position the printline horizontally. It can be relative or absolute placement. If relative, the position is from the current position. If absolute, it is measured from the left margin of the logical page.

SAME Specifies that the vertical position and type (absolute or relative) of the printline is the same as the **previous coded** XLAYOUT command, if the previously coded XLAYOUT has an absolute-x POSITION. Otherwise the x-position will be changed to "RELATIVE 0" causing it to print at the current position.

If this is the first XLAYOUT of the PAGEFORMAT the print position is at the left margin as defined in the PAGEFORMAT or PAGEDEF command.

Note: SAME is not recommended on an XLAYOUT if the previous XLAYOUT has an absolute-x POSITION, because the behaviour of changing the POSI-

TION to "RELATIVE 0" is not guaranteed to be the same in future releases of PPFA.

vert Specifies a number of units (inches, cm, mm, etc) used to position the printline vertically. It can be relative or absolute placement. If relative the position is from the current position. If absolute it is measured from the top margin.

XML example data & pagedef

Below is a simple example of XML data and the page definition to print it.

```
<customer type='Home'>
  <name>
    <first>Johnny</first>
    <last>Johnson</last>
  </name>
  <address>
    <strno>234</strno>
    <street>Wickerfield  Road</street>
    <city>Zeni</city>
    <state>KYKentucky</state>
    <zip>42214</zip>
  </address>
</customer>

<customer type='Work'>
  <name>
    <first>JR</first>
    <last>Horn</last>
  </name>
  <address>
    <strno>140</strno>
    <street>Main  Street</street>
    <city>Nothington</city>
    <state>KYKentucky</state>
    <zip>71504</zip>
  </address>
</customer>
```

Using the following page definition and the above data, I want to print:

```
Home customer: Johnny Johnson      234 Wickerfield Road
                                   Zeni, KY 42214

Work customer: JR Horn             100 Main Street
                                   Nothington, KY 71504
```

```
SETUNITS 1 IN 1 IN LINESP 6 LPI;
Pagedef XMLxml1 replace yes UDType EBCDIC;
 FONT   MYFONT TYPE EBCDIC;
 DEFINE cust     QTAG 'customer';
 DEFINE name     QTAG 'customer','name';
 DEFINE fname    QTAG 'customer','name','first';
 DEFINE lname    QTAG 'customer','name','last';
 DEFINE addr     QTAG 'customer','address';
 DEFINE strno    QTAG 'customer','address','strno';
 DEFINE street   QTAG 'customer','address','street';
 DEFINE city     QTAG 'customer','address','city';
 DEFINE state    QTAG 'customer','address','state';
 DEFINE zip      QTAG 'customer','address','zip';

 XLAYOUT cust    POSITION ABSOLUTE 0   Relative 0.333;
   FIELD ATTR 'type' ;
   FIELD TEXT ' customer:' ;
 XLAYOUT fname   POSITION ABSOLUTE 2.5      SAME;
 XLAYOUT lname   POSITION RELATIVE 0.167    SAME;
 XLAYOUT strno   POSITION ABSOLUTE 5.5      SAME;
 XLAYOUT street POSITION RELATIVE 0         SAME;
   FIELD TEXT ' '         ;
   FIELD START 1 LENGTH *;
 XLAYOUT city    POSITION ABSOLUTE 5.5     NEXT;
   FIELD START 1 LENGTH *;
   FIELD TEXT   ', ';
 XLAYOUT state   POSITION RELATIVE 0       SAME;
   FIELD START 1 LENGTH 2;
   FIELD TEXT   ' ';
 XLAYOUT zip     POSITION RELATIVE 0       SAME;
```

The Font command

identifies a font that is to be used subsequently in a command placing text. Prior to adding the XML data support to page definitions, PPFA assumed that the font encoding and the data encoding were single byte EBCDIC, and translated the text to EBCDIC if necessary (ie if PPFA is invoked on an ASCII system). PPFA allowed the font encoding to be specified but it was used only for generating page numbers for record formatting page definitions.

Page definitions that support XML data must specify the font encoding and it will be used to translate the text if it differs from the data encoding. The data encoding is specified on the PAGEDEF command using the UDType subcommand. When the font encoding does not agree with the data encoding, the data must be transformed to match that of the font. PPFA simply passes that information in the page definition. The transformation is done outside PPFA. Only the following transformations are allowed and then only if they are supported by the presentation device:

- UDType UTF-8, Font type can only be ASCII or UNICODE.
- UDType UTF-16, Font type can only be UNICODE.

If the Font type differs from the User Data Type and it is any other combination PPFA will issue an error message and no page definition will be created.

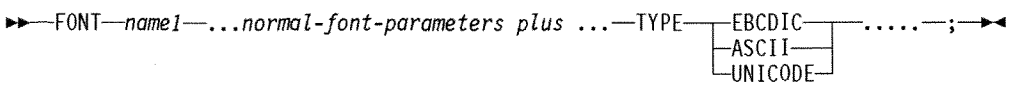

Figure 4. FONT changed syntax drawing

TYPE Specifies the encoding type of a font. This parameter is required when defining fonts for XML page definitions.

EBCDIC Single byte EBCDIC code page 500.

ASCII Single byte ASCII code page 819.

UNICODE (or UCS2) Fixed two byte unicode without surrogates.

The Field command

The Field command parses an XML data item. The only change in field command syntax is that RECID is now STAG meaning the actual start tag name is to be printed.

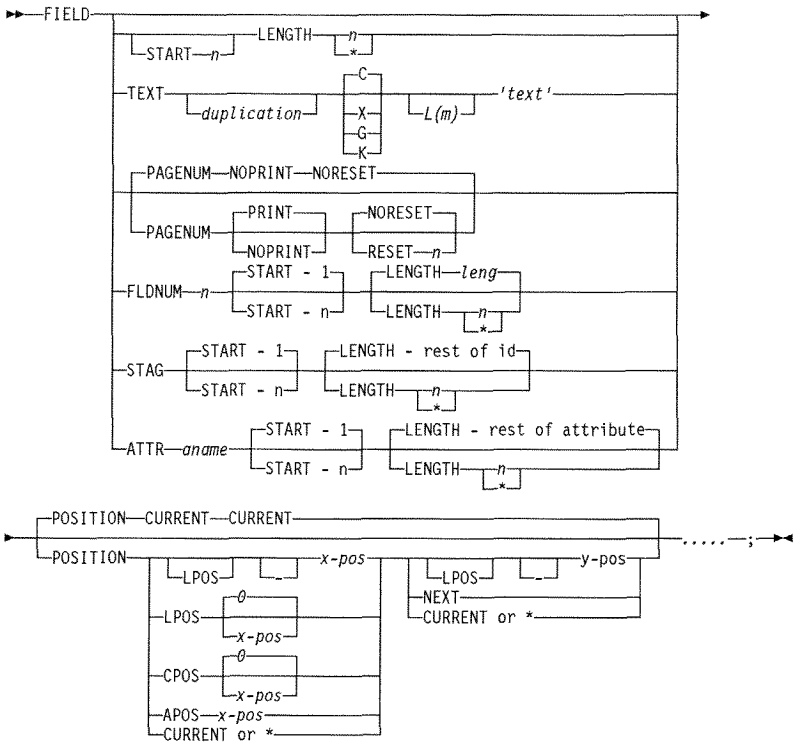

Figure 5. syntax of modified FIELD command

Note: When the XLAYOUT command has relative-x positioning the following restrictions apply to the FIELD command:

- FIELD commands with x-positioning relative to the XLAYOUT (LPOS).
- FIELD ATTR (attribute) with x-positioning relative to the XLAYOUT (LPOS).
- FIELD commands with barcodes.

START/LENGTH & TEXT These parameters are the same as prior to these enhancements. LENGTH * means the rest of the field.

PAGENUM n Although both parameters are specified as optional, at least one must be specified. Page numbers could be set at this point to start with the value specified as n, however they normally will follow the specification made in the Pagedef or Pageformat command. The POSITION parameters specified with the PAGENUM parameter will reflect the position of the page number only. If you do not wish a page number printed, either do not use this parameter or specify NOPRINT.

The RESET parameter is only used when you wish to reset the page number that is to be used starting with this page.

FLDNUM This keyword should only be used if the DELIMITER field was used in the XLAYOUT command. Fields cannot be counted without delimiters being specified in the database.

To allow for the identification of a part of a field which has been numbered, you can specify the starting position (from the delimiter) and the length of the field to be used. If START isn't coded, 1 is used. If LENGTH isn't coded the remainder of the field is used.

STAG This keyword allows you to access characters in the start tag itself. This also includes the "<" and ">" delimiters, so that position 1 is always the "<".

If no record length is specified, the remaining bytes of the start tag is assumed. If no START is specified, 1 is assumed.

ATTR This keyword allows you to access attribute values from the data. Multiple attribute fields can access the same attribute thus allowing subsets of the value to be printed.

If no record length is specified, the remaining bytes of the attribute are assumed. If no START is specified, 1 is assumed.

aname The attribute name. If you want to preserve the case put the name in quotes. This name will be converted to the users' data type as specified, using UDTYPE on the page definition, or defaulted.

START n The starting position of the attribute to extract the data. If this parameter is omitted, position 1 is assumed.

Length n The length of the attribute to be placed. If this parameter is omitted, the rest of the attribute field is assumed.

POSITION Specifies the starting position or the alignment position of the field in the printout.

Notes:

1. When POSITION is not coded, both the x-position and y-position default to CURRENT.
2. LPOS in the x-position is not allowed when the x-position of the associated XLAYOUT is relative.

x-position The x-axis or Inline positioning parameters. These parameters allow for positioning relative both to the containing XLAYOUT (LPOS) and to the current position (CPOS and CURRENT). However, positioning relative to the XLAYOUT is currently restricted to XLAYOUTs with absolute-x positioning.

LPOS Specifies that this parameter is relative to the XLAYOUT position.

Note: This parameter is not allowed when its' associated XLAYOUT has relative-x positioning.

x-pos Optional inline positioning offset. This parameter can be negative, and when omitted defaults to 0.

CPOS Specifies that this parameter is relative to the current position.

x-pos Optional inline positioning offset. This parameter can be negative, and when omitted defaults to 0.

APOS Specifies that the x-pos parameter that follows is absolute.

x-pos Mandatory inline positioning offset. This parameter must be positive.

CURRENT or * Specifies that the inline offset (relative to the fields direction) is the end of the previous field. For the first field, use the XLAYOUT offset. If POSITION is not coded, this is the default.

Note: This parameter is "grandfathered". You can get the same results by simply coding CPOS.

Note: All x-pos offsets are specified by an up to 3 decimal place number followed, optionally, by the units. The unit choices are **IN**, **MM**, **CM**, **POINTS** or **PELS**.

y-position The y-axis or Baseline positioning parameters. These parameters allow for positioning relative both to the XLAYOUT (LPOS) position.

LPOS Specifies that this parameter is relative to the XLAYOUT position. This is the default.

- ((((use the existing words for this parameter))))

y-pos ((((use the existing words for this parameter))))

NEXT ((((use the existing words for this parameter))))

CURRENT ((((use the existing words for this parameter))))

***** ((((use the existing words for this parameter))))