Proposed Hitachi XML Layout

Here are my current thoughts. A full example appears at the end of the document.

For messages going to the printer, values are only sent if they appear in the input XML file and they reflect a value that is not the “Default” setting. Otherwise they are ignored.

When retrieving messages from the printer, an attempt is made to include only referenced parameters and only those that do not reflect the default setting. E.G., An offset of 0, a Substitution setting of Disable.

Root Class

The root would be a “Label”.

```
[XmlRoot("Label", IsNullable = false)]
public class Lab {
   [XmlAttribute]
   public string Version;   // Keep track of version to allow for changes
   public Printer Printer;  // Information that pertains to the printer
   public Msg Message;      // Information that pertains to the message
}
```

Printer Class

The printer section covers the printer setup for printing a message. It is broken into subsections (just because I did it that way for cijConnect). They may need to be collapsed into the Printer Class.

```
public class Printer {
   [XmlAttribute]
   public string Make;
   [XmlAttribute]
   public string Model;
   public PrintHead PrintHead;
   public ContinuousPrinting ContinuousPrinting;
   public TargetSensor TargetSensor;
   public CharacterSize CharacterSize;
   public PrintStartDelay PrintStartDelay;
   public EncoderSettings EncoderSettings;
   public InkStream InkStream;
   public Substitution Substitution;
   public Logos Logos;
}
```

When moving from printer to printer, the print head orientation may change.

```
public class PrintHead {
   [XmlAttribute]
   public string Orientation;
}
```

No idea how often this is used.

```
public class ContinuousPrinting {
   [XmlAttribute]
   public string RepeatInterval;
   [XmlAttribute]
   public string PrintsPerTrigger;
```

```
}
```

No idea how often this is used.

```
public class TargetSensor {
    [XmlAttribute]
    public string Filter;
    [XmlAttribute]
    public string SetupValue;
    [XmlAttribute]
    public string Timer;
}
```

Width can only be sent for Time Based printing

```
public class CharacterSize {
    [XmlAttribute]
    public string Height;
    [XmlAttribute]
    public string Width;
}
```

Do not know the rules for setting Reverse Delay.

```
public class PrintStartDelay {
    [XmlAttribute]
    public string Forward;
    [XmlAttribute]
    public string Reverse;
}
```

Divisor is only used for encoder based printing. Maybe width should be here

```
public class EncoderSettings {
    [XmlAttribute]
    public string HighSpeedPrinting;
    [XmlAttribute]
    public string Divisor;
    [XmlAttribute]
    public string ExternalEncoder;
}
```

I think there are more settings like Interlaced and Single Scan.

```
public class InkStream {
    [XmlAttribute]
    public string InkDropUse;
    [XmlAttribute]
    public string ChargeRule;
}
```

The intent here is to be aple to carry along all logos that are used by the message. Specifying the folder would allow the handled to get a fresh copy of the logo. There is nowhere in the printer to store the folder name.

```
public class Logos {
    [XmlAttribute]
    public string Folder;
    [XmlElement("Logo")]
    public Logo[] Logo;
}
```

Substitution and Substitution Rule are a work-in-progress

```
public class Substitution {
   [XmlAttribute]
   public string Delimiter;
   [XmlAttribute]
   public string StartYear;
   [XmlAttribute]
   public string Rule;

   [XmlElement("Rule")]
   public SubstitutionRule[] SubRule;

}

public class SubstitutionRule {
   [XmlAttribute]
   public string Type;
   [XmlAttribute]
   public string Base;

   public string Text;
}
```

The layout and location allow messages to be sent to the printer. If a messahe is retriened from the printer, the bitmaps would be stored here.

```
public class Logo {
   [XmlAttribute]
   public string Layout;
   [XmlAttribute]
   public string Location;
   [XmlAttribute]
   public string RawData;
   [XmlAttribute]
   public string FileName;
}
```

Message Class

The message class describes the message. This layout only describes the “Individual” lauout. It may have to be expanded for “Overall” layout. It definitely will need to be expanded for “Free” Layout.

The current implementation uses the “Add Column” to allocate a column and “Line Count” to set the number of items in the column. The Column Number and Item Number is assigned by the printer so it is not included in the specification of a column or item. There can be multiple columns in a message.

```
public class Msg {
   [XmlAttribute]
   public string Layout;
   [XmlElement("Column")]
   public Column[] Column;
}
```

A column is made up of multiple items.

```
public class Column {
   [XmlAttribute]
   public string InterLineSpacing;
   [XmlElement("Item")]
   public Item[] Item;
}
```

A item contains all the information stored in the printer to descibe one item. Date and Counter Classes are Arrays to handle the case where multiple calendar and count are included in a single text string. It might be a good idea to allow multiple Text Classes to handle very long strings. .The Location objects is used for internal processing but is not included in the XML since all the values are assigned by the printer and not by the use. The data and counter cannot appear in the same item.

```
public class Item {
   [XmlAttribute]
   public string Type;
   public FontDef Font;
   public BarCode BarCode;
   [XmlElement("Date")]
   public Date[] Date;
   [XmlElement("Counter")]
   public Counter[] Counter;
   public string Text;

   [XmlIgnore]
   public Location Location;
}
```

The Location objects is used for internal processing but is not included in the XML since all the values are assigned by the printer and not by the user.

```
public class Location {
   public int Row;     // 0-Origin
   public int Col;     // 0-Origin
   public int Index;   // 0-Origin
   public int X;       // 0-Origin
   public int Y;       // 0-Origin
   public int calStart = 0;
   public int calCount = 0;
   public int countStart = 0;
   public int countCount = 0;
}
```

The Font class describes the unique characteristics of the font.

```
public class FontDef {
   [XmlAttribute]
   public string IncreasedWidth;
   [XmlAttribute]
   public string InterCharacterSpace;
   [XmlAttribute]
   public string DotMatrix;
}
```

The Barcode Class is a work in progress. Maybe include it with Font Class?

```
public class BarCode {
   [XmlAttribute]
   public string HumanReadableFont;
   [XmlAttribute]
   public string EANPrefix;
   [XmlAttribute]
   public string DotMatrix;
}
```

Counter Class

The counter class is an object within an item. It was arbitrarily broken into sub-classes. All open for discussion.

```
public class Counter {
   [XmlAttribute]
   public int Block;

   public Range Range;
   public Count Count;
   public Reset Reset;
   public Misc Misc;
}

public class Range {
   [XmlAttribute]
   public string Range1;
   [XmlAttribute]
   public string Range2;
   [XmlAttribute]
   public string JumpFrom;
   [XmlAttribute]
   public string JumpTo;
}

public class Count {
   [XmlAttribute]
   public string InitialValue;
   [XmlAttribute]
   public string Increment;
   [XmlAttribute]
   public string Direction;
   [XmlAttribute]
   public string ZeroSuppression;
}

public class Reset {
   [XmlAttribute]
   public string Type;
   [XmlAttribute]
   public string Value;
}
```

```
public class Misc {
   [XmlAttribute]
   public string UpdateUnit;
   [XmlAttribute]
   public string UpdateIP;
   [XmlAttribute]
   public string Multiplier;
   [XmlAttribute]
   public string ExternalCount;
   [XmlAttribute]
   public string CountSkip;
}
```

Date Class

The Date class is an object within an item. It was arbitrarily broken into sun-classes. All open for discussion.

```
public class Date {
   [XmlAttribute]
   public int Block;
   [XmlAttribute]
   public string SubstitutionRule;
   [XmlAttribute]
   public string RuleName;
   public Offset Offset;
   public ZeroSuppress ZeroSuppress;
   public Substitute Substitute;
   public TimeCount TimeCount;
   [XmlElement("Shift")]
   public Shift[] Shift;
}

public class Offset {
   [XmlAttribute]
   public string Year;
   [XmlAttribute]
   public string Month;
   [XmlAttribute]
   public string Day;
   [XmlAttribute]
   public string Hour;
   [XmlAttribute]
   public string Minute;
}
```

```
public class ZeroSuppress {
   [XmlAttribute]
   public string Year;
   [XmlAttribute]
   public string Month;
   [XmlAttribute]
   public string Day;
   [XmlAttribute]
   public string Hour;
   [XmlAttribute]
   public string Minute;
   [XmlAttribute]
   public string Week;
   [XmlAttribute]
   public string DayOfWeek;
}

public class Substitute {
   [XmlAttribute]
   public string Year;
   [XmlAttribute]
   public string Month;
   [XmlAttribute]
   public string Day;
   [XmlAttribute]
   public string Hour;
   [XmlAttribute]
   public string Minute;
   [XmlAttribute]
   public string Week;
   [XmlAttribute]
   public string DayOfWeek;
}
```

Shift was included as a sub-section of the Date Class since it results in a Calendat Object being generated..

```
public class Shift {
   [XmlAttribute]
   public int ShiftNumber;
   [XmlAttribute]
   public string StartHour;
   [XmlAttribute]
   public string StartMinute;
   [XmlAttribute]
   public string EndHour;
   [XmlAttribute]
   public string EndMinute;
   [XmlAttribute]
   public string ShiftCode;
}
```

Time Count was included as a sub-section of the Date Class since it results in a Calendar Object being generated..

```
public class TimeCount {
   [XmlAttribute]
   public string Interval;
   [XmlAttribute]
   public string Start;
   [XmlAttribute]
   public string End;
   [XmlAttribute]
   public string ResetTime;
   [XmlAttribute]
   public string ResetValue;
}
```

A Complete(?) Example

A test that should contain an example of everything.

```
<?xml version="1.0" encoding="utf-8"?>
<Label Version="1">
  <Printer Make="Hitachi" Model="UX-D161W">
    <PrintHead Orientation="Inverted/Forward" />
    <ContinuousPrinting RepeatInterval="0" PrintsPerTrigger="1" />
    <TargetSensor Filter="Until End of Print" SetupValue="50" Timer="0" />
    <CharacterSize Height="90" Width="10" />
    <PrintStartDelay Forward="96" Reverse="96" />
    <EncoderSettings HighSpeedPrinting="HM" Divisor="1" ExternalEncoder="None" />
    <InkStream InkDropUse="2" ChargeRule="Standard" />
    <Substitution Delimiter="/" StartYear="2019" Rule="1">
      <Rule Type="Month" Base="1">JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT</Substitution>
      <Rule Type="Month" Base="11">NOV/DEC</Substitution>
      <Rule Type="DayOfWeek" Base="1">MON/TUE/WED/THU/FRI/SAT/SUN</Substitution>
    </Substitution>    <Logos Folder="">
      <Logo Layout="Fixed" Location="" RawData="" />
    </Logos>
  </Printer>
  <Message Layout="Individual">
    <Column InterLineSpacing="2">
      <Item Type="Date">
        <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
        <BarCode />
        <Date Block="1" SubstitutionRule="1" RuleName="">
          <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
          <ZeroSuppress Year="Disable" Month="Disable" Day="Space Fill" />
          <Substitute Month="Enable" />
        </Date>
        <Text>SELL BY {{MMM}/{DD}/{YY}}  </Text>
      </Item>
      <Item Type="Date">
        <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
        <BarCode />
        <Date Block="1" SubstitutionRule="1" RuleName="">
          <Offset Day="20" />
          <ZeroSuppress Year="Disable" Month="Disable" Day="Space Fill" />
          <Substitute Year="Disable" Month="Disable" Day="Disable" />
        </Date>
        <Text>USE BY  {{MMM}/{DD}/{YY}}  </Text>
      </Item>
      <Item Type="Date">
        <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
        <BarCode />
        <Date Block="1" SubstitutionRule="1" RuleName="">
          <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
          <ZeroSuppress DayOfWeek="Disable" />
          <Substitute DayOfWeek="Enable" />
        </Date>
```

```
      <Text>PACKED  {{TTT} {777}} </Text>
    </Item>
  </Column>
  <Column InterLineSpacing="2">
    <Item Type="Date">
      <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
      <BarCode />
      <Date Block="1">
        <Shift ShiftNumber="1" StartHour="0" StartMinute="0" EndHour="7" EndMinute="59" ShiftCode="D" />
        <Shift ShiftNumber="2" StartHour="8" StartMinute="0" EndHour="15" EndMinute="59" ShiftCode="E" />
        <Shift ShiftNumber="3" StartHour="16" StartMinute="0" EndHour="23" EndMinute="59" ShiftCode="F" />
      </Date>
      <Text>Shift {{E}}</Text>
    </Item>
    <Item Type="Calendar">
      <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
      <BarCode />
      <Date Block="1">
        <TimeCount Interval="30 Minutes" Start="A1" End="X2" ResetTime="6" ResetValue="A1" />
      </Date>
      <Text>TCount {{FF}} </Text>
    </Item>
    <Item Type="Counter">
      <Font IncreasedWidth="1" InterCharacterSpace="1" DotMatrix="5x8(5x7)" />
      <BarCode />
      <Counter Block="1">
        <Range Range1="000000" Range2="999999" JumpFrom="000199" JumpTo="000300" />
        <Count InitialValue="000001" Increment="2" Direction="Down" ZeroSuppression="Enable" />
        <Reset Type="Signal 1" Value="000001" />
        <Misc UpdateUnit="1" UpdateIP="0" Multiplier="" />
      </Counter>
      <Text># {{CCCCCC}} </Text>
    </Item>
  </Column>
  <Column InterLineSpacing="0">
    <Item Type="Text">
      <Font IncreasedWidth="1" InterCharacterSpace="2" DotMatrix="18x24" />
      <BarCode />
      <Text>{X/000}</Text>
    </Item>
  </Column>
 </Message>
</Label>
```

Current Thoughts on Printer Composition

Sections I am wondering about:

1. Substitution Class

The intent here is to be able to reload the substitution files in the printer. Probably need to add the Name of the File.

The Delimiter cannot be stored and retrieved from the printer so may need to be fixed.

The Start Year was placed as an attribute. Need a rational place to put it.

The Rule Class layout is just to allow the something like Month to be subdivided.

2. Logo Class

Needs lots of thought here. Need to see how the printer responds to reading a free layout logo.

3. Printer Class

The Printer Class is optional. If it is missing, no settings will change in the printer. The sub-divisions are arbitrary. More thought required. Things like Time Based vs. Encoder based.

Current Thoughts on Message Composition

Sections I am wondering about:

1. Message Class

The Message Class is optional. This can be used just to load the Substitution Rules.

The Message Class only specifies Layout (Free, Overall, Individual). It contains the collection of Columns. If the Message Class is included, there must be at least one column within the message. Within the printer, there is always at least one Column and one Item.

2. Column Class

Column Class only specifies Inter-line Spacing. You cannot specify a Column Number since the printer assigns column numbers

3. Item Class.

Item Class currently specifies a Type (Text, Date, or Counter) but that is also assigned by the printer based on the text in the message. That should be removed.

The Item Class always contains a Font Class, A Barcode Class, and a Text Class. It might work to move the Font Information into the Item Class as Attributes.

The Barcode Class could stay as it is or be moved into the Item Class as attributes.

The Text Class needs to be defined as a collection so very long text messages could be loaded. Care must be taken on breaking the text into sections as that can cause messages text to be discarded in the printer.

4. Date Class

The Date Class is broken into five sub-classes. The Date Class is a collection since multiple Calendar Blocks can be assigned to a single Item Class.

The Block Number is a 1-origin index of the Calendar Blocks assigned to this Item Class. The Calendar Starting Block and the number of blocks is assigned by the printer.

5. Shift Class

The Shift Class has an entry for each shift. The Shift Class is currently a collection. Should it be moved into another Sub-class?

6. Counter Class

The Counter Class is was arbitrarily broken into sub-Classes. I do not use them enough to know what makes sense.