

 Seamless

 Integrated

 Consistent

 Personalised

**enactor**<sup>®</sup>  
retail systems for a digital world

# Enactor Training Course

## Editing Receipt Documents

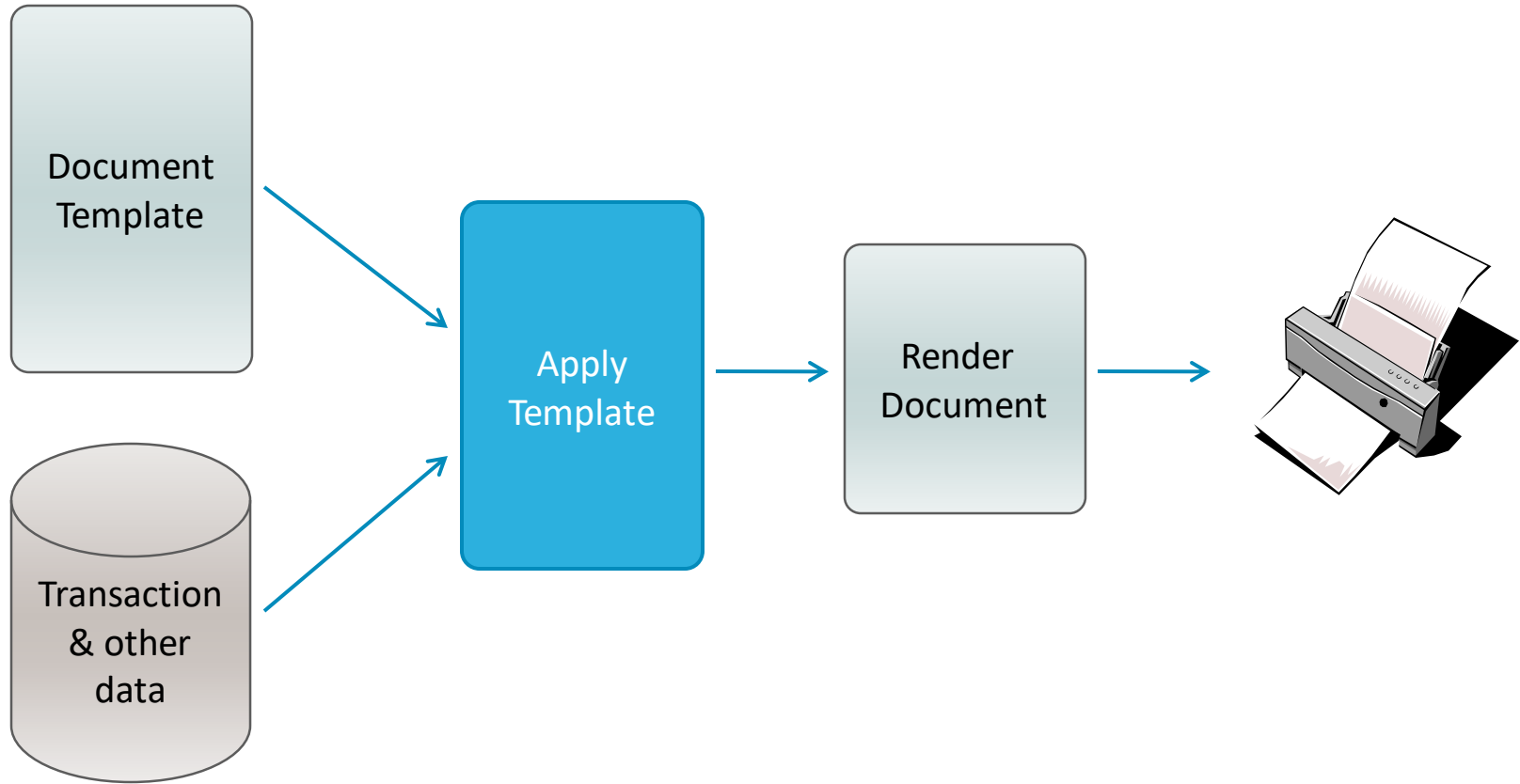


## Receipt Documents

# Document Templates

- All POS Receipts and Slips are formatted using Document Templates
- A Document Template is an XML file describing how to layout the text
  - Uses tags vaguely similar to HTML, but is a proprietary format
- Templates are also used to format some output for display
  - Items in Lists
  - Items for the line display

# Print Process



# Template Content

- A snippet of XML from a document template:

```
<core:column align="CENTRE" columnDefId="0">  
  <core:elements>  
    <core:lineElement messageId="SALE_HEADER" colour="blue"  
      style="BOLD;DOUBLE_HIGH;DOUBLE_WIDE"/>  
  </core:elements>  
</core:column>
```

# Template Content (cont.)

- Simple Output Tags:
  - Text, LineElement, Barcode, Image, Page
- Structure Tags:
  - Row, RowLine
  - Column, ColumnDef
  - TableTemplate
  - FormatEntry, FormatElement
- Data Manipulation Tags:
  - EntityReference, Assign

# EL Expressions

- Print templates use EL (Expression Language) to access data:

```
<core:text value="One Street"/>
```

Outputs the text "One Street"

```
<core:text value="#{location.locationAddress.street1}"/>
```

Assumes that a data item "location" is accessible and has a property called "locationAddress" which in turn has a property called "street1" which is a printable string.

# EL Expressions (cont.)

- Expressions can include functions

```
<core:line visibility="#{notEmpty(location.locationAddress.street2)}">
<core:elements>
  <core:column align="LEFT" columnDefId="AddressDetail">
    <core:elements>
      <core:text
value="#{location.locationAddress.street2}" />
    </core:elements>
  </core:column>
</core:elements>
</core:line>
```



# Text Formatting

- Formatting can be applied to text elements using a formatter:

```
<core:text value="#{item.foreignAmount}">  
  <core:formatters>  
    <core:formatCurrencyAmount  
currencyId="#{item.currencyId}" />  
  </core:formatters>  
</core:text>
```

## Text Formatter (cont.)

- Or you can include formatting within the text itself:

```
<core:text messageId="AMOUNT_DUE"/>
```

- Or within the text message resource:

```
<core:message key="AMOUNT_DUE">The amount due is  
{format:formatCurrencyCurrency(tenderAmount, tender.currencyId)}.  
</core:message>
```

# Format Entries

- Common blocks of output which are going to be re-used are grouped into Format Entries
- Format Entries are referred to by Format Elements
- Format Elements have “applicability” expressions
- Format Elements are typically used within tables (lists)
  - When the applicability matches, the associated Format Entry content is output
- ‘Visibility’ also applies when not in tables.

# Table Templates

- Table Templates are used to perform iteration over a list
- A list expression defines the list, e.g.
  - `<core:tableTemplate list="#{basket.nonTenderItemIterator}" id="Items">`
- Within the table template, format entries are used to bring in associated content. For each data item in the list, the first matching entry is used

```
<core:formatElement  
  applicableClassName="com.enactor.mfc.basket.items.IMerchandiseItem"  
  applicability="#{abs(item.quantity) <= 1 && !item.voided}"  
  formatId="merchandiseItem" />
```

# Included Templates

- An outer template can “include” others
- Included templates are searched when locating Format Entries
- Allows the main print templates to be built using a selection of common Format Entries
- Outer or prior included template can “override” included Format Entries

# Included Templates (cont.)

- Enactor provide a number of standard includes:
  - Pos/Print/ReceiptCommonColumnDefsNN
  - Pos/Print/ReceiptItemFormatEntries
  - Pos/Print/ReceiptGeneralFormatEntriesNN
  - Where NN is the print width
- These help provide a default format for all the built-in items in Enactor

# Print Document Editor

- There is a designer to help with print document templates
- In practice tend to need to use both the designer and edit the XML directly
- Designer doesn't currently offer help with data expressions, though validation is applied.

# Configuration of Receipts

- Pos Terminal Maintenance, Printing tab
- Select from pre-defined Templates
  - Get a default in most cases if not specified
- The selections are defined in Page Definition Detail Maintenance
  - Just encapsulate the Document Template URL
  - A standard set is installed by default



# Configuration of Receipts (cont.)

- When you setup a a new template, you need to also setup a Page Definition Details as well so that it can be selected in the Maintenance UI
- Pos Terminal includes a selection for Receipt List
  - Used by the operator receipt display to format each basket item as an individual object.

 Seamless

 Integrated

 Consistent

 Personalised

**enactor**<sup>®</sup>  
retail systems for a digital world

**Q & A**