

Work with InterSystems.
Not separate systems.



ZEN Reports

Steve Pisani

INTERSYSTEMS

Agenda



- **Background**
 - Why develop ZEN Reports ?
 - Technologies adopted: XML, XSLT and XPATH
 - ZEN Reporting process.
- **ZEN Report class anatomy**
 - XData Report Definition: Building an XML data source
 - XData ReportDisplay : Building and styling a ZEN Report layout
 - Sample ZEN Report #1
 - ZEN Report Templates
 - Report Styling, XSL Templates and extending XSL.
- **PDF Rendering Tools**
 - Troubleshooting PDF Rendering
- **ZEN Report Class parameters**
- **Generating a ZEN report from the command line**
- **Sample Reports**
- **ZEN Reports New and Next**

INTERSYSTEMS

Why the work on ZEN Reports ?



- Why is this effort important ?
 - All Customers have some need to do reporting, from legacy to state-of-the art
 - Dominated by Crystal which is hugely expensive.
- We can supply an alternative to Crystal.
- Used internally by HealthShare, TrakCare and DeepSee.

INTERSYSTEMS

Background ZEN Report technology



- XML Extensible Markup Language
- XSLT Extensible Stylesheet Language Transformation
- XPATH XML Path Language

INTERSYSTEMS

XML - Extensible Markup Language



- XML File showing XML declaration, Root Tags, Elements, Attributes, Value and Contents

```
<?xml version="1.0" encoding="utf-8" ?>
<SalesReport runTime="2009-02-04 11:56:50" month="ALL">
  <SalesRep name="Jack Bradsmith">
    <record id="68" number="4">
      <date>2005-04-09</date>
      <customer>PicoPlex LLC.</customer>
    </record>
    <record id="55" number="6">
      <date>2005-04-12</date>
      <customer>MegaCalc Partners</customer>
    </record>
  </SalesRep>
  <SalesRep name="Jack Bradsmith">
    ...
  </SalesRep>
</SalesReport>
```

INTERSYSTEMS

XSLT - Extensible Stylesheet Language Transformation



- Sample XSLT language embedded within HTML and XML

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl='http://www.w3.org/1999/XSL/Transform' >
  <xsl:template match="/">
    <html>
      <body>
        <xsl:apply-templates select='/SemesterReport/UniversityDetails' />
        <h1>Grades for Semester <xsl:value-of
          select='/SemesterReport/SemesterDetails/Semester' /></h1>
        <hr />
        <p />
        <xsl:apply-templates
          select='SemesterReport/SemesterDetails/Classes/Class' />
      </body>
    </html>
  </xsl:template>
  <SemesterReport>
    <UniversityDetails>...
    ...
  </SemesterReport>
</xsl:stylesheet>
```

INTERSYSTEMS

XPATH - XML Path Language



- XPath is a language for finding information in an XML document. It is used to navigate through elements and attributes in an XML document.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book><book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. Note: JES and later has implemented that [0] should be the first node, but according to the W3C standard it should have been [1]!
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element.
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element.
/bookstore/book[position() < 3]	Selects the first two book elements that are children of the bookstore element.
//title[@lang]	Selects all the title elements that have an attribute named lang.
//title[@lang="eng"]	Selects all the title elements that have an attribute named lang with a value of 'eng'.
/bookstore/book[price > 35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00.
/bookstore/book[price > 35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00.

INTERSYSTEMS

Source XML + XSL = Target XML



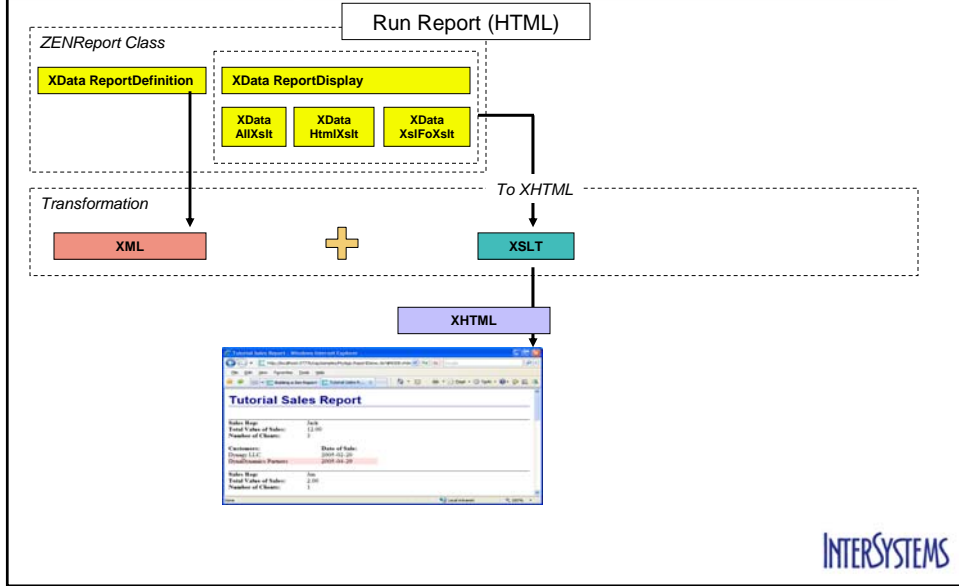
```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet
  type="text/xsl" href="helloworld.xsl"?>
<doc>
  <title>Hello World Test</title>
  <hello>World</hello>
</doc>
```

```
<html>
<head>
  <title>Hello World Test</title>
</head>
<body>
  <p>Hello World</p>
</body>
</html>
```

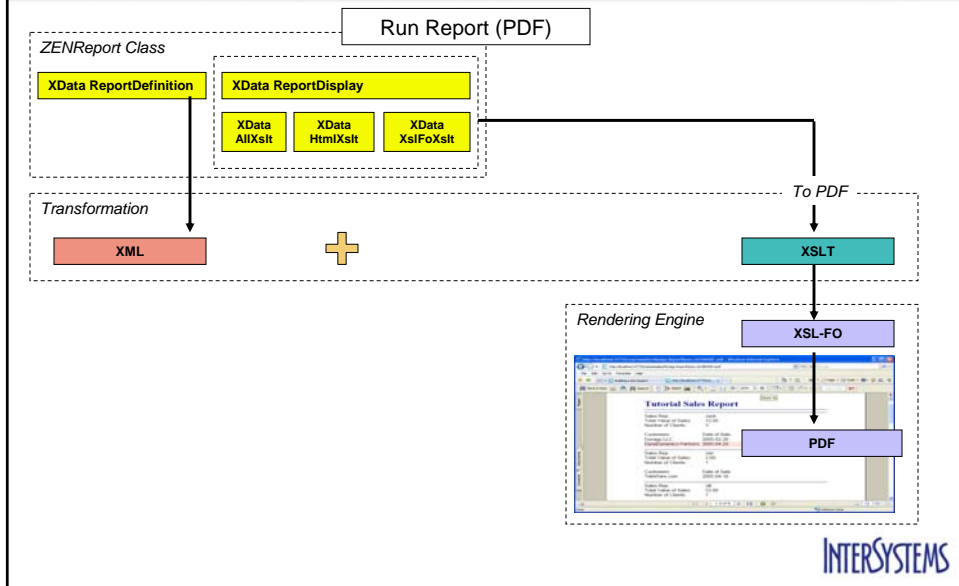
```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" encoding="iso-8859-1" indent="yes" />
  <xsl:template match="/">
    <html> <head>
      <title><xsl:value-of select="doc/title"/></title> </head>
      <body>
        <p>Hello <xsl:value-of select="doc/hello"/></p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

INTERSYSTEMS

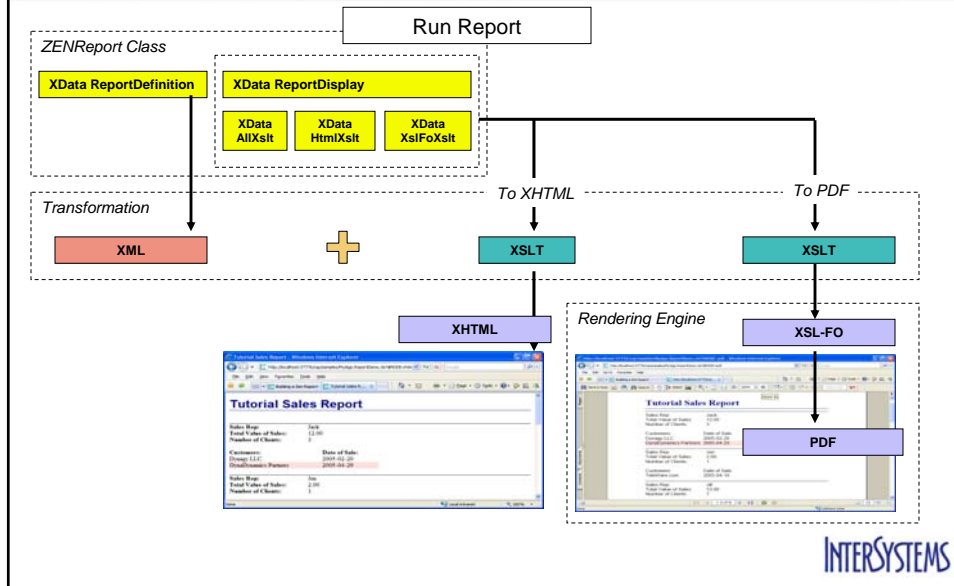
ZENReport Process animation (HTML)



ZEN Report Process animation (PDF)



ZEN Report Process diagram



INTERSYSTEMS

ZEN Report Class makeup



- Class ***ClassName*** Extends %ZEN.Report.reportPage
[parameters...]
...for configuring the generation and execution of the report.
- XData ReportDefinition {
...for generating XML document 'source'
 }
- XData ReportDisplay {
...describing the output layout
(generating XSL used for an HTML or an PDF target)
 }

INTERSYSTEMS

XData ReportDefinition



- XData ReportDefinition specifies data acquisition and XML formatting of that data

```
XData ReportDefinition [XMLNamespace="http://www.intersystems.com/zen/report/definition"]
{
<report xmlns="http://www.intersystems.com/zen/report/definition"
  sql="SELECT TOP 5 Name,SSN,Age FROM Sample.Person ORDER BY Age">

<attribute name="attName" expression="$Username" />
<attribute name="classname" expression="..%ClassName(1)" />

  <group name="Person">
    <attribute name="FullName" field="Name"/>
    <attribute name="SSN " field="SSN"/>
    <element name="Age" field="Age" />
  </group>
</report>
}
```

INTERSYSTEMS

XData ReportDefinition



- The data supplied for the report can be retrieved by specifying:
 - SQL String
 - Caché Query
 - Class Method that returns a result set.
- An pre-defined XML file containing the XML formatted data can also be supplied, omitting the need for the XData ReportDefinition block. (see DATASOURCE)

INTERSYSTEMS

orderby Attribute



- The **orderby** attribute of the <report> or <group> element allows you to override any *ORDER BY* SQL clause supplied.

This allows you to supply a URI parameter for re-sorting your report.

```
Property SortOrder As %String(ZENURL="SortOrder")  
[InitialExpression="SalesRep,SaleDate"];
```

```
<report xmlns="http://www.intersystems.com/zen/report/definition"  
sql="SELECT ID, Customer, Num, SalesRep, SaleDate  
FROM ZENApp_Report.Invoice  
WHERE (Month(SaleDate) = ?) OR (? IS NULL)  
ORDER BY SalesRep, SaleDate" orderby="!..SortOrder" >
```

INTERSYSTEMS

<group> and *breakOn** attributes



- Create nested XML elements using <group> and the *breakOnField* (or *breakOnExpression*) attribute of <group>:

Invoice	Item	Cost
1000	Shirt	40.00
1000	Shoes	90.00
1001	Shirt	45.00
1001	Pants	99.50
1001	Shoes	26.30
1002	Shirt	43.20
...		

```
<report xmlns="http://www.intersystems.com/zen/report/definition"  
sql="Select Invoice,Item,Cost from SQLUser.InvoiceItem  
ORDER BY Invoice">  
  <group name="Invoice" breakOnField="Invoice">  
    <attribute name="InvoiceNumber" field="Invoice"/>  
    <group name="Items">  
      <element name="Cost" field="Cost"/>  
      <element name="Item" field="Item" />  
    </group>  
  </group>  
</report>
```

INTERSYSTEMS

<group> and *breakOn** attributes



- Create nested XML elements using <group> and the *breakOnField* (or *breakOnExpression*) attribute of <group>:

Invoice	Item	Cost
1000	Shirt	40.00
1000	Shoes	
1001	Shirt	
1001	Pants	
1001	Shoes	
1002	Shirt	
...		

```

<report xmlns="http://www.intersystems.com/zen/report/definition"
  sql="Select Invoice,Item,Cost from SQLUser.InvoiceItem
  ORDER BY Invoice">
  <?xml version="1.0" encoding="utf-8" ?>
  <report>
    <Invoice InvoiceNumber="1000">
      <Items>
        <Cost>40.00</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    <Invoice InvoiceNumber="1001">
      <Items>
        <Cost>45.00</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    <Invoice InvoiceNumber="1002">
      <Items>
        <Cost>99.50</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    .....
  </report>
  
```

INTERSYSTEMS

Aggregates



- Create XML elements that represent aggregates (SUM,AVG,CUSTOM...) using <aggregate> :

```

...
<group name="Invoice" breakOnField="Invoice">
  <attribute name="InvoiceNumber" value="1000" />
  <aggregate name="TotalCost" />
  <group name="Items">
    <element name="Cost" />
    <element name="Item" />
  </group>
</group>
...
  <?xml version="1.0" encoding="utf-8" ?>
  <report>
    <Invoice InvoiceNumber="1000">
      <TotalCost>130.00</TotalCost>
      <Items>
        <Cost>40.00</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    <Invoice InvoiceNumber="1001">
      <TotalCost>213.00</TotalCost>
      <Items>
        <Cost>45.00</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    <Invoice InvoiceNumber="1002">
      <TotalCost>99.50</TotalCost>
      <Items>
        <Cost>99.50</Cost>
        <Item>Shirt</Item>
      </Items>
    </Invoice>
    .....
  </report>
  
```

EMMS

XData ReportDisplay



- XData ReportDisplay defines the report *layout*. An XSL stylesheet is generated

```
XData ReportDisplay [ XMLNamespace = "http://www.in
...
<report xmlns="http://www.intersystems.com/zen/report/
...
<document width="8.5in" height="11in" ..... />
...
<body>
...
<group name="Invoice">
  <item field="@InvoiceNumber" caption=" Invoice
  <table group="Items" orient="col" altcolor="#DEEF
    <item field="Item" width="2in" caption="Item Purc
    <item field="Cost" width="2in" caption="Amount"/>
    ...
  </table>
  ...
</group>
...
</body>
</report>
}
```

- <init> — XSL parameters to pass to the
- <document> — general page characteristics for PDF output
 - ... <class> — for defining style classes
- <pageheader>, <pagefooter> — page headers/ footers content

- <barChart>, <lineChart>, <pieChart> — Charts
- <block> — Group of items for a table row or column
- <header>, <footer> — Containers for propagating styles
- <group> — Group of items for repeated actions
- — Image
- <inline> — Inline text styling
- <item> — Data value
- <line> — Horizontal line
- <list> — Bulleted or numbered list
- <p> — Text string of any length
- <pagebreak> — Page break
- <table> — Table
- <fo>, <html>, <write>....

INTERSYSTEMS

Sample ZEN Report



- Open dc.Invoices.cls
 - XData ReportDefinition contents
 - XData ReportDisplay contents
- Run the report in HTML and/or PDF

INTERSYSTEMS

ZEN Report Templates



- ZEN Report Templates, are typically used for common headers/footers.
- Usage:

- Create Class extending:

```
%ZEN.Report.Display.reportTemplate
```

- Call from any element using the template attribute:

```
<header template="dc.Templates.myHDR:CompanyA"/>
```

INTERSYSTEMS

Formatting Report Display CSS styles



- A standard set of CSS styles are supplied for reports

```
<report ... style="standard/none" .../>
```

- p.banner1
- table.table1, ... table.table6
- td
- td.table1, ... td.table6
- th.grid
- th.table1, ... th.table6

Invoice Number: 2	
Purchased Item	Amount
Pants	41.00
Shirt	95.00
Shoes	82.00
Shirt	69.00
Shoes	60.00
Shoes	26.00

INTERSYSTEMS

Overriding default CSS styles



- The `<class>` element (under `<document>` in XDATA ReportDefinition) renders CSS class definitions into the XHTML report, and into equivalent XSL stylesheet information for the PDF report.

```
<document ...>
<class name="th.SalesList">
  <att name="background-color" value="blue" />
  <att name="color" value="white" />
</class>
</document>
```

```
...
<table orient="col" class="SalesList">
  <item field="@name" width="6in"/>
  ...
</table>
```

Name = **tagName.className**

Where **tagName** may be the name of one of the following HTML tags only:

table, td, th, a, inline or p

The **className** portion can be any chosen name that uniquely identifies this style. Eg:

```
name="th.SalesList"
name="td.myTable"
```

INTERSYSTEMS

<cssinclude> and <xslinclude>



- The `<cssinclude>` and `<xslinclude>` tags are used in XHTML and PDF output respectively.
- External file content referenced by these tags will be included in the HTML and XSL-FO stylesheets respectively - as is.

```
<document ....>
  <cssinclude href="myStyle.css" />
  <xslinclude href="myStyle.xsl" />
</document>
```

```
th.myTable {
  background-color: blue;
  color: white;
}
```

myStyle.css

```
<xsl:attribute-set name="th.myTable">
  <xsl:attribute name="background-color">
    blue</xsl:attribute>
  <xsl:attribute name="color">
    white</xsl:attribute>
</xsl:attribute-set>
```

myStyle.xsl

INTERSYSTEMS

Display element *style* attribute .



- Specific styles can be associated with individual display items:

```
<item special="number" width="75%" style="padding-right:4px;color:yellow"/>
```

- Or conditionally defined:

```
<item special="number" width="75%"  
  selectstylecond = "%report.Month=1,1"  
  selectstylelist = "border:none;padding-right:4px;color:red,padding-  
    right:4px;color:yellow"  
>
```

Note: As evaluation of server-side variables in expressions (*selectstylecond*) requires XSLT transformation that executes on the server, hence the report parameter XSLTMODE must be equal to 'server' (\$XSLT)

INTERSYSTEMS

XSL Templates to apply styles



- XSL templates can be defined in the Zen Report class to be applied based on different output formats. (2009.*)

```
XData HtmlXslt {  
  <xsl:template name="redeven" >  
    <xsl:param name="num"/>  
    <xsl:if test="$num mod 2 = 0">  
      <xsl:attribute name="style" >color:red</xsl:attribute>  
    </xsl:if>  
  </xsl:template>  
}
```

```
XData XsIFoXslt { ... }
```

```
XData AllXslt { ... }
```

```
XData ReportDisplay [ XMLNamespace = "http://www.intersystems.com/zen/report/display" ]  
{  
  ...  
  <item field="@id" width=".7in" style="border:none;padding-right:4px"  
    stylecall="redeven" styleparams="@id" styleparamNames="num" >  
  ...  
}
```

Note: This type of conditional style evaluation does not need XSLTMODE='server' set.

INTERSYSTEMS

Extending XSLT support: <xslt>



- <xslt> element enables you to contribute XSL instructions (defined within the Class) to a Zen report. (2009. *)

```
XData setsize {  
<xsl:variable name="height">  
<xsl:choose>  
<xsl:when test="$orientation='landscape'">  
<xsl:value-of select="210mm"/>  
</xsl:when>  
<xsl:otherwise>  
<xsl:value-of select="297mm"/>  
...  
...  
}
```

```
XData ReportDisplay [ XMLNamespace = "http://www.intersystems.com/zen/report/display" ]  
{  
  <report xmlns="http://www.intersystems.com/zen/report/display" name="Container">  
    ...  
    <xslt all="setsize" vars="orientation" expressions="!%report.Orientation"/>  
    ...  
  </report>  
}
```

INTERSYSTEMS

Conditional ZEN Report Display items



- Every XData ReportDisplay element except <document> allows its output to be controlled using conditional "if" expressions.

(ifexpression, ifxpath, includeCollfExpression and includeColUnlessExpression)

Eg:

```
<table ifexpression='$ROLES["Supervisor"]' ..... >
```

Note: As evaluation of server-side variables in expressions requires XSLT transformation that executes on the server, hence the report parameter XSLTMODE must be equal to '**server**' (\$XSLT)

INTERSYSTEMS

Rendering Tools



- The **Rendering Tools** take the XML document that is the product of the XML to XSL-FO translation, generates a PDF document.
- **Rendering Tools** are third party products.
Eg: Open-Source APACHE product called 'FOP'
A RenderX product called 'XEP'
- This operation requires %System_CallOut:USE privilege
- Refer Caché to the batch or script file that invokes the engine by setting a %SYS global:

```
Set ^%SYS("zenreport","transformerpath")="C:\fop-0.95\fop.bat"
```

INTERSYSTEMS

Troubleshooting PDF Generation



- Intermediate files – URI parameters
 - \$MODE - view different outputs
 - \$LOG - return log file of transformations
 - \$NODELETE - retain intermediate files
(requires \$LOG=1 for XHTML)
 - \$REPORTNAME - rename intermediate files
- Hints on solving PDF Errors
 - .../fop -xml *test.xml* -xsl *test.xsl* -pdf *test.pdf*

INTERSYSTEMS

ZEN Report class parameters



- **DEFAULTMODE** (*\$MODE*)
 - Identifies the default output mode for the report
Possible values are: 'xml', 'tohtml', 'html', 'toxslfo', 'xslfo' and 'pdf'
- **DATASOURCE** (*\$DATASOURCE*)
 - Identifies an XML document that contains the data for the Zen report. The DATASOURCE value can be the URI of any valid XML document. Relative URIs are handled with respect to the current URI.
- **HTMLSTYLESHEET** (*bypassing XDATA ReportDefinition in HTML mode*)
 - Identifies the **to-HTML stylesheet** that controls XHTML output for the Zen report. This string can be the URI of any to-HTML valid XSL stylesheet. Relative URIs are handled with respect to the current URI. - Replaces the use of the XDATA ReportDefinition block for XHTML output
- **XSLFOSTYLESHEET** (*bypassing XDATA ReportDefinition in PDF mode*)
 - Identifies the **to-XSLFO stylesheet** that controls PDF output for the Zen report. This string can be the URI of any valid XSL stylesheet. - Replaces the use of the XDATA ReportDefinition block for PDF output

Note: Parameters represented here as (*\$parameter*) can be specified as URI values

INTERSYSTEMS

ZEN Report class parameters (cont)



- **USETEMPFILES**
 - When USETEMPFILES=1 (true) Zen writes its generated XSL stylesheets for HTML consumption to file. It subsequently uses the generated XSL stylesheet in the file to generate the output XHTML. The default for USETEMPFILES is 0 (false). In this case Zen generates and uses XSL but does not save it to a file. (*\$USETEMPFILES*)
- **EMBEDXSL**
 - (*\$EMBEDXSL*) Specifies whether XSLT is embedded with XML, if it is we must use explicit namespaces for generated XML by specifying values for **REPORTXMLNAMESPACE**, **REPORTXMLNAMESPACEPREFIX**
- **REPORTNAME**
 - Controls the filename suggested by the browser when you ask to save the final output of running a Zen report in XHTML or PDF format. If you do not supply a value for the REPORTNAME, Zen uses the report class name as the filename.

Note: Parameters represented here as (*\$parameter*) can be specified as URI values

INTERSYSTEMS

Features: ZEN Report Images



- Images
 - Use the element in the XData ReportDefinition block to insert an image into the report.

```

```

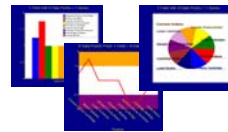
- If the report is going to be output to PDF too, then specify *contentHeight* and *contentWidth* attributes.

INTERSYSTEMS

Features: ZEN Report Charts



- Charts are incorporated using <*chart> tags in the XData ReportDefinition block
 - Line, Bar and Pie Charts supported



```
<pieChart title="1 Field with 9 Data Points = 1 Series"  
dataFields="!@AdultPrice"  
seriesGroup="Theater"  
seriesNames="!@TheaterName"  
seriesColors="blue,red,green,pink,orange,plum,purple,brown,navy"  
  
width="450px" height="400px"  
marginTop="10" marginLeft="15" marginRight="5"  
  
legendVisible="false" />
```

INTERSYSTEMS

Creating Reports from the command line



- Setup a %CSP.Request object for the request,
- Invoke **GenerateReport**(<path to save report>,<mode>)

```
Zn "SAMPLES"

Set %request=##class(%CSP.Request).%New()
Set %request.URL = "/csp/samples/ZENApp.ZenReportImageExample.cls"
Set %request.CgiEnvs("SERVER_NAME")="127.0.0.1"
Set %request.CgiEnvs("SERVER_PORT")=57774

Set report=##class(ZENApp.ZenReportImageExample).%New()
Set Status=report.GenerateReport("C:\CacheSys\CSP\Samples\Reps\X.PDF",2)

If 'Status Do DecomposeStatus^%apiOBJ(Status,.Err) w !,Err(Err)
Write !,Status
```

INTERSYSTEMS

ZEN Report Sample #2



- Sample reports dc.Invoices2 including:
 - Images
 - Report Templates
 - Charts

Other Samples: Preserve space report
 Label Report

INTERSYSTEMS

ZEN Reports – New and Next...



- Other recent 2008.2 and 2009.1 enhancements ...:
 - Easy specification and support for page orientation.
 - Removing and/or Suppressing empty XML nodes from reports
 - Suppressing repetition of columns and rows in a table
 - Number formatting using formatNumber attribute
 - <inline> tag for mixing styles on the same line of text
 - Improved performance
 - Aggregates define in report Display as well as Definition.
- Some of what we are working on...
 - ZEN Report Generator
 - ZEN Report Front-end based on DynaForm
 - Support for pre-printed Forms
 - and more...

INTERSYSTEMS

Work with InterSystems.
Not separate systems.



ZEN Reports

Steve Pisani

INTERSYSTEMS