XSLT Commands

February 2013/rm

censhare

Overview

censhare

- Definitions of Wikipedia
  - XSLT
    - XSLT (E**x**tensible **S**tylesheet **L**anguage **T**ransformation) is a declarative, XML-based language used for the transformation of XML documents.

Overview

censhare

- censhare
  - censhare has own implementation of XML, XPath 2.0 and XSLT 2.0
    - Advantages
      - XML implementation is faster and has less objects in memory
      - XPath and XSLT implementation gives expandability without 3rd party license costs (like Saxon) and is faster
      - Additional XPath functions, e.g. asset query, additional xsl instructions (e.g. xml update) will follow
    - Disadvantages
      - Not the entire range is implemented
      - List of supported operators and functions is available
    - Documented functionality
      - This training shows only the supported functionality of the censhare implementation of version 4.7

censhare XSLT enhancements

censhare

- Instructions
  - We have implemented own instructions to enhance XSLT
  - All censhare functions have the namespace „cs“ for „**c**en**s**hare“. This namespace is not optional and must exist

censhare XSLT enhancements

censhare

- Instructions
  - XSLT commands
    - Structure
      - <cs:command name="[name of the command]" returning="[result variable]">
        <cs:param name="[name of the parameter]" select="[value of the parameter]"/>
        <cs:param name="[name of the parameter]" select="[value of the parameter]"/>
        </cs:command>

censhare XSLT enhancements

censhare

- Instructions
  - XSLT commands
    - Results can be collected in 3 ways
      - Using the „returning“ attribute
        - <xsl:variable name="resultAssetXml"/>
          <cs:command name="com.censhare.api.assetmanagement.CloneAndCleanAssetXml" returning="resultAssetXml">
            <cs:param name="source" select="asset"/>
          </cs:command>
      - Using the result of the „cs:command“ function
        - <cs:command name="com.censhare.api.assetmanagement.CloneAndCleanAssetXml">
            <cs:param name="source" select="asset"/>
          </cs:command>
      - Using the „dest“ parameter (not available at every command)
        - <cs:command name="com.censhare.api.io.WriteXML">
            <cs:param name="source" select="$metaDoc"/>
            <cs:param name="dest" select="concat($out, 'EPUB-Debug.xml')"/>
          </cs:command>

XSLT Commands

censhare

- Asset management
  - Commands
    - Check in new asset
    - Check out asset
    - Check in asset
    - Check out abort asset
    - Update asset
    - Clone and clean asset XML

XSLT Commands

censhare

- Asset management
  - Check in new asset command
    - Create asset of given asset XML
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.CheckInNew“
      - „returning“: XML of created asset
    - Parameters
      - „source“: asset XML to create

XSLT Commands

censhare

- Asset management
  - Check in new asset command
    - Example
      - <xsl:variable name="assetXml">
        <asset name="First test asset" type="temp."/>
        </xsl:variable>
        <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CheckInNew" returning="resultAssetXml">
        <cs:param name="source" select="$assetXml"/>
        </cs:command>
      - <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CheckInNew" returning="resultAssetXml">
        <cs:param name="source">
        <asset name="First test asset" type="temp."/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Asset management
  - Check out asset command
    - Check out asset with given asset XML or ID
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.CheckOut“
      - „returning“: XML of checked out asset
    - Parameters
      - „source“: asset XML or asset ID to checkout
    - Example
      - ```
        <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CheckOut" returning="resultAssetXml">
         <cs:param name="source">
          <asset id="211421" currversion="0"/>
         </cs:param>
        </cs:command>
        ```
        ```

XSLT Commands

censhare

- Asset management
  - Check in asset command
    - Check in asset with given asset XML or ID
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.CheckIn“
      - „returning“: XML of checked in asset
    - Parameters
      - „source“: asset XML or asset ID to checkin
    - Example
      - ```
        <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CheckIn" returning="resultAssetXml">
          <cs:param name="source">
            <asset id="211421" currversion="0"/>
          </cs:param>
        </cs:command>
        ```
        ```

XSLT Commands

censhare

- Asset management
  - Check out abort asset command
    - Check out abort asset with given asset XML or ID
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.CheckOutAbort“
      - „returning“: XML of checked out abort asset
    - Parameters
      - „source“: asset XML or asset ID to check out abort
    - Example
      - ```
        <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CheckOutAbort" returning="resultAssetXml">
          <cs:param name="source">
            <asset id="211421"/>
          </cs:param>
        </cs:command>
        ```
        ```

XSLT Commands

censhare

- Asset management
  - Update asset command
    - Update asset with given asset XML (update in same asset version)
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.Update“
      - „returning“: XML of updated asset
    - Parameters
      - „source“: complete asset XML to update

XSLT Commands

censhare

- Asset management
  - Update asset command
    - Example
      - <xsl:variable name="sourceAssetXml">
        <xsl:copy-of select="asset"/>
        </xsl:variable>
        <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.Update" returning="resultAssetXml">
        <cs:param name="source">
        <asset>
        <xsl:copy-of select="$sourceAssetXml/asset/@*"/>
        <xsl:attribute name="name" select="'Test'"/>
        <xsl:copy-of select="$sourceAssetXml/asset/node()"/>
        </asset>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Asset management
  - Clone and clean asset XML command
    - Clone and clean asset XML with given asset XML or ID (can be used to check in new asset)
    - Attributes
      - „name“: „ com.censhare.api.assetmanagement.CloneAndCleanAssetXml“
      - „returning“: XML of cloned and cleaned asset
    - Parameters
      - „source“: complete asset XML to clone and clean
    - Example
      - <xsl:variable name="resultAssetXml"/>
        <cs:command name="com.censhare.api.assetmanagement.CloneAndCleanAssetXml" returning="resultAssetXml">
        <cs:param name="source" select="asset"/>
        </cs:command>

XSLT Commands

censhare

- Asset management
  - Example: add selected asset to basket
    - <!-- get basket -->
      <xsl:variable name="basketAsset" select="cs:asset()[@censhare:asset.type='order.shoppingbasket.'
      and @censhare:asset.wf_target=system-property('censhare:party-id')]"/>
      <!-- create relation to basketAsset -->
      <xsl:if test="$basketAsset and empty($basketAsset/child_asset_rel[@key='user.' and @child_asset=current()/@id])">
      <xsl:variable name="resultAssetXml"/>
      <cs:command name="com.censhare.api.assetmanagement.Update" returning="resultAssetXml">
      <cs:param name="source">
      <asset>
      <xsl:copy-of select="$basketAsset/@* | $basketAsset/node()"/>
      <child_asset_rel key="user." child_asset="{@id}"/>
      </asset>
      </cs:param>
      </cs:command>
      </xsl:if>

XSLT Commands

censhare

- Asset management
  - Example: remove selected asset from basket
    -
```
<!-- get basket -->
<xsl:variable name="basketAsset" select="cs:asset()[@censhare:asset.type='order.shoppingbasket.'
  and @censhare:asset.wf_target=system-property('censhare:party-id')]"/>
<!-- remove relation from basketAsset -->
<xsl:if test="$basketAsset">
  <xsl:variable name="resultAssetXml"/>
    <cs:command name="com.censhare.api.assetmanagement.Update" returning="resultAssetXml">
      <cs:param name="source">
       <asset>
         <xsl:copy-of select="$basketAsset/@* | $basketAsset/node()[not(node-name(.)='child_asset_rel' and @key='user.'
           and @child_asset=current()/@id)]"/>
       </asset>
    </cs:param>
  </cs:command>
</xsl:if>
```

XSLT Commands

censhare

- Transformation
  - Commands
    - XSL transformation
    - Asset transformation
    - Image transformation
    - Barcode transformation
    - FOP transformation

XSLT Commands

censhare

- Transformation
  - XSL transformation
    - Execute a XSL transformation (sub transformation)
    - Attributes
      - „name“: „com.censhare.api.transformation.XslTransformation“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input XML to be transformed as XML, URL, file locator or storage item (required)
      - „dest“: The file locator of the destination file where the result of the transformation should be stored (optional)
        If this parameter is missing, the result is returned as result of this cs:command function
      - „stylesheet“: File locator to the stylesheet file (required)
      - „transform“: An XML node with additional parameters, which is passed to the XSL stylesheet without further interpretation (optional)

XSLT Commands

censhare

- Transformation
  - XSL transformation
    - Example
      - <cs:command name="com.censhare.api.transformation.XslTransformation" returning="dummy">
        <cs:param name="stylesheet" select="concat($config, 'modules/content_export/stylesheets/export.xsl'"/>
        <cs:param name="source" select="$chapterAsset/storage_item[@key='master'][1]"/>
        <cs:param name="dest" select="concat($parentDir, 'chapter-', $chapterAsset/@id, '.xhtml')"/>
        <cs:param name="transform">
        <transform format="xhtml" debug="true"/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Transformation
  - Asset transformation
    - Executes an asset transformation (sub transformation)
    - Same as „XSL transformation“, but stylesheet is used from a resource asset
    - Attributes
      - „name“: „com.censhare.api.transformation.AssetTransformation“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input XML to be transformed as XML, URL, file locator or storage item (required)
      - „dest“: The file locator of the destination file where the result of the transformation should be stored (optional)
        If this parameter is missing, the result is returned as result of this cs:command function
      - „key“: Key to the stylesheet resource asset (required)
      - „transform“: An XML node with additional parameters, which is passed to the XSL stylesheet without further interpretation (optional)

XSLT Commands

censhare

- Transformation
  - Asset transformation
    - Example
      - <cs:command name="com.censhare.api.transformation.AssetTransformation" returning="dummy">
        <cs:param name="key" select="'censhare:epub-chapter'"/>
        <cs:param name="source" select="$chapterAsset/storage_item[@key='master'][1]"/>
        <cs:param name="dest" select="concat($parentDir, 'chapter-', $chapterAsset/@id, '.xhtml')"/>
        <cs:param name="transform">
        <transform format="xhtml"/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Transformation
  - Image transformation
    - Executes one or more image transformations (e.g. crop, scale, …)
    - Attributes
      - „name“: „com.censhare.api.transformation.ImageTransformation“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input source (image) to be transformed (required). It must reference a storage item or a file
      - „dest“: The file locator of the destination file where the result of the transformation should be stored (required). It must reference a file in a virtual file system. The file extension is used to define the target image format (e.g. „a.jpg“)
      - „image-transformation“: XML node, which contains the image transformations and transformation attributes (required)

XSLT Commands

censhare

- Transformation
  - Image transformation
    - Operations
      - Scale: <scale width="220" height="220"/>
        Crop: <crop x="100" y="20" width="220" height="220"/>
        Rotate: <rotate degree="90"/>
        Convert: <convert/>
        OPI: <opi/>
        Flip horizontally: <fliph/>
        Flip vertically: <flipv/>
        Overlay: <overlay overlay-image="cs:file-locator($overlayAsset/storage_item[@key='master'])" opacity="50" gravity="center" tile="false" extra-opts=""/>
        Video frame: <videoframe timecode="0:0:0.000"/>
        Clip (Use an embedded clipping path to produce a new image with a transparent alpha channel): <clip/>
        Frame (Scale input image proporionally to fit into a frame of a given width and height): <frame upscale="false" width="100" height="100" gravity="center" clip="false"/>

XSLT Commands

censhare

- Transformation
  - Image transformation
    - Example
      -
```
<cs:command name="com.censhare.api.transformation.ImageTransformation">
  <cs:param name="source" select="$currentAsset/storage_item[@key='master']"/>
  <cs:param name="dest" select="concat($dir, 'imageTransformation/image.jpg')"/>
  <cs:param name="image-transformation">
    <image-transformation>
      <operations>
        <crop x="100" y="20" width="220" height="220"/>
        <rotate degree="90"/>
        <clip/>
      </operations>
      <attributes sharpen="true" out-jpeg-quality="100" out-color="cmyk"/>
    </image-transformation>
  </cs:param>
</cs:command>
```

XSLT Commands

censhare

- Transformation
  - Barcode transformation
    - Executes a barcode transformation
    - Used fonts must be available at startup of the JavaVirtualMachine
    - Attributes
      - „name“: „com.censhare.api.transformation.BarcodeTransformation“
    - Parameters
      - „source“: The input value to be represented by the barcode
      - „dest“: The file locator of the destination file where the result of the transformation should be stored. It must reference a file in a virtual file system (required)
      - „barcode-transformation“: XML node, which contains the barcode transformations (required)

XSLT Commands

censhare

- Transformation
  - Barcode transformation
    - Parameter
      - „type“: ean8, ean13, qr (further types possible but not implemented yet)
        „file-format“: pdf, eps, svg, png
        „height“: mm value
        „module-width“: space between „bars“, leave empty for an automatic default value
        „margin“: mm value
        „orientation“: 0, 90, 180, 270 (degrees rotation)
        „font“: font name must be available to the JVM at server startup time
        „font-size“: pt value
        „qr-error-correction“: „“, „H“, „L“, „M“, „Q“
        „dpi“: dots per inch for rasterized image formats
        „antialias“: „true“, „false“ for font antialiasing

XSLT Commands

censhare

- Transformation
  - Barcode transformation
    - Example
      - <cs:command name="com.censhare.api.transformation.BarcodeTransformation">
        <cs:param name="source" select="$source"/>
        <cs:param name="dest" select="$resultFile"/>
        <cs:param name="barcode-transformation">
        <barcode-transformation type="ean13" file-format="pdf" height="30" module-width="1" margin="4"
        orientation="0" font="Courier" font-size="6" qr-error-corr="" dpi="300" antialias="true"/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Transformation
  - FOP transformation
    - Executes a FOP transformation (sub transformation)
    - Attributes
      - „name“: „com.censhare.api.transformation.FopTransformation“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input XML to be transformed as XML, URL, file locator or storage item (required)
      - „dest“: The file locator of the destination file where the generated PDF should be stored. It must reference a file in a virtual file system (optional)
      - „key“: Key to the stylesheet resource asset (required)
      - „transform“: An XML node with additional parameters, which is passed to the XSL stylesheet without further interpretation (optional). The given node will be available within the transformation via the named parameter „transform“. The attributes author, title, pdf-base-url, keywords, source-resolution, target-resolution and pdf-base14-kerning are passed to the FOP processor

XSLT Commands

censhare

- Transformation
  - FOP transformation
    - Example
      - <cs:command name="com.censhare.api.transformation.FopTransformation" returning="dummy">
        <cs:param name="key" select="'censhare:fop-sample'"/>
        <cs:param name="source" select="asset/storage_item[@key='master'][1]"/>
        <cs:param name="dest" select="concat($parentDir, 'chapter-', $chapterAsset/@id, '.pdf')"/>
        <cs:param name="transform">
        <transform target-resolution="300"/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- Office
  - Commands
    - Read Excel
    - Read Word

XSLT Commands

censhare

- Office
  - Read Excel
    - Read Excel document as XML
    - Attributes
      - „name“: „com.censhare.api.Office.ReadExcel“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input XML to be transformed as XML, URL, file locator or storage item (required)
      - „dest“: The file locator of the destination file. It must reference a file in a virtual file system (optional)
      - „max-columns“: Maximal number of columns to read (optional, default is unlimited)
      - „max-rows“: Maximal number of rows to read (optional, default is unlimited)
      - „level“: „info“ reads only information about sheets, styles and print areas. „full“ reads all (optional, default is „full“)
      - „sheet-index“: reads only sheet with given index (optional)

XSLT Commands

censhare

- Office
  - Read Excel
    - Example
      - <cs:command name="com.censhare.api.Office.ReadExcel">
        <cs:param name="source" select="asset/storage_item[@key='master'][1]">
        </cs:command>

XSLT Commands

censhare

- Office
  - Read Word
    - Read Word document as XML
    - Attributes
      - „name“: „com.censhare.api.Office.ReadWord“
      - „returning“: Result of the transformation (optional)
    - Parameters
      - „source“: The input XML to be transformed as XML, URL, file locator or storage item (required)
      - „dest“: The file locator of the destination file. It must reference a file in a virtual file system (optional)

XSLT Commands

censhare

- Office
  - Read Word
    - Example
      - <cs:command name="com.censhare.api.Office.ReadWord">
        <cs:param name="source" select="asset/storage_item[@key='master'][1]">
        </cs:command>

XSLT Commands

censhare

- Context
  - Commands
    - Set property

XSLT Commands

censhare

- Context
  - Set property
    - Set static properties (variables) in global context
    - Attributes
      - „name“: „com.censhare.api.context.SetProperty“
    - Parameters
      - „name“: Name of the property with values
        - „censhare:logger“: ???
        - „censhare:command-xml“: ???
        - „censhare:working-dir“: ???
        - „censhare:auto-delete“: set auto delete of working directory to „true“ or „false“
        - „censhare:result-file-locator“: set result file
        - „censhare:result-mimetype“: set result mime type
      - „value“: Value of the property

XSLT Commands

censhare

- Context
  - Set property
    - Example
      - Disable deletion of the working directory at end of transaction
        - <cs:command name="com.censhare.api.context.SetProperty">
          <cs:param name="name" select="'censhare:auto-delete'"/>
          <cs:param name="value" select="'false'"/>
          </cs:command>
      - Set result file and result mimetype
        - <cs:command name="com.censhare.api.context.SetProperty">
          <cs:param name="name" select="'censhare:result-file-locator'"/>
          <cs:param name="value" select="$destFile"/>
          </cs:command>
          <cs:command name="com.censhare.api.context.SetProperty">
          <cs:param name="name" select="'censhare:result-mimetype'"/>
          <cs:param name="value" select="$sourceStorage/@mimetype"/>
          </cs:command>

XSLT Commands

censhare

- I/O
  - Commands
    - Write XML
    - Read XML
    - Copy
    - Create virtual file system
    - Close virtual file system
    - Get file system reference

XSLT Commands

censhare

- I/O
  - Write XML
    - Writes XML to a file
    - Attributes
      - „name“: „com.censhare.api.io.WriteXML“
    - Parameters
      - „source“: The XML to be written (required)
      - „dest“: The file locator of the destination file (required). It must reference a file in a virtual file system
      - „output“ An XML element „output“ providing various serialization properties like pretty printing or encoding (optional). This element is identical to the XSLT output function (xsl:output) and supports most of it‘s attributes

XSLT Commands

censhare

- I/O
  - Write XML
    - Parameter
      - name? = qname
        method? = "xml" | "html" | "xhtml" | "text" | qname-but-not-ncname
        byte-order-mark? = "yes" | "no"
        cdata-section-elements? = qnames
        doctype-public? = string
        doctype-system? = string
        encoding? = string
        escape-uri-attributes? = "yes" | "no"
        include-content-type? = "yes" | "no"
        indent? = "yes" | "no"
        media-type? = string
        normalization-form? = "NFC" | "NFD" | "NFKC" | "NFKD" | "fully-normalized" | "none" | nmtoken
        omit-xml-declaration? = "yes" | "no"
        standalone? = "yes" | "no" | "omit"
        undeclare-prefixes? = "yes" | "no"
        version? = nmtoken

XSLT Commands

censhare

- I/O
  - Write XML
    - Example
      - <cs:command name="com.censhare.api.io.WriteXML">
        <cs:param name="source" select="$metaDoc"/>
        <cs:param name="dest" select="concat($out, 'EPUB-Debug.xml')"/>
        <cs:param name="output">
        <output indent="yes"/>
        </cs:param>
        </cs:command>

XSLT Commands

censhare

- I/O
  - Read XML
    - Reads XML content from a file or a storage item
    - Attributes
      - „name“: „com.censhare.api.io.ReadXML“
      - „returning“: readed XML (optional)
    - Parameters
      - „source“: The input source where to read from (file or storage item)

XSLT Commands

censhare

- I/O
  - Read XML
    - Example
      - <cs:command name="com.censhare.api.io.ReadXML" returning="content">
        <cs:param name="source" select="asset/storage_item[@key='master'][1]"/>
        </cs:command>

XSLT Commands

censhare

- I/O
  - Copy
    - Copies a file or a storage item into a destination file
    - Attributes
      - „name“: „com.censhare.api.io.Copy“
    - Parameters
      - „source“: The input source to be copied (file or storage item)
      - „dest“: The file locator of the destination file. It must reference a file in a virtual file system

XSLT Commands

censhare

- I/O
  - Copy
    - Example
      - Copy a storage item
        - <cs:command name="com.censhare.api.io.Copy">
          <cs:param name="source" select="asset/storage_item[@key='master'][1]"/>
          <cs:param name="dest" select="concat($out, 'result/image.jpg')"/>
          </cs:command>
      - Copy a file
        - <cs:command name="com.censhare.api.io.GetFilesystemRef" returning="config"/>
          <cs:param name="name" select="'config-runtime'"/>
          </cs:command>
          <cs:command name="com.censhare.api.io.Copy">
          <cs:param name="source" select="concat($config, 'modules/resources/logo.jpg'"/>
          <cs:param name="dest" select="concat($out, 'result/image.jpg')"/>
          </cs:command>

XSLT Commands

censhare

- I/O
  - Create virtual file system
    - Creates a (possibly virtual) directory, which can be used by the caller to write files into it
    - Attributes
      - „name“: „com.censhare.api.io.CreateVirtualFileSystem“
      - „returning“: The URL identifying the virtual file system. This string can (and must) be used for all subsequent output file references
    - Parameters
      - „format“: The result format of the file system (optional). If the file system should be packed into an archive, this parameter must be specified. Currently the only supported format is ZIP (value "zip"). If the parameter is missing, the result will be a directory structure

XSLT Commands

censhare

- I/O
  - Create virtual file system
    - Example
      - <cs:command name="com.censhare.api.io.CreateVirtualFileSystem" returning="out"/>
        <cs:param name="format" select="'zip'"/>
        </cs:command>

XSLT Commands

censhare

- I/O
  - Close virtual file system
    - Closes a virtual file system
    - Attributes
      - „name“: „com.censhare.api.io.CloseVirtualFileSystem“
      - „returning“: The URL identifying the resulting directory or archive file. It can be used as a „censhare:result-file-locator“ return parameter
    - Parameters
      - „id“: The file system reference as returned in „CreateVirtualFileSystem“
      - „name“: The name of the result archive file (optional). This parameter is ignored if the file system is either no ZIP archive or the parameter „dest“ is specified
      - „dest“: A file reference where the archive file should be stored (optional). This parameter is ignored if the file system isn’t an archive

XSLT Commands

censhare

- I/O
  - Close virtual file system
    - Example
      - ```
        <cs:command name="com.censhare.api.io.CreateVirtualFileSystem" returning="out"/>
          <cs:param name="format" select="'zip'"/>
        </cs:command>
        ...
        <cs:command name="com.censhare.api.io.CloseVirtualFileSystem" returning="zip">
          <cs:param name="id" select="$out"/>
          <cs:param name="name" select="concat(asset/@name, '.zip')"/>
        </cs:command>
        ```
        ```

XSLT Commands

censhare

- I/O
  - Get file system reference
    - Get a reference to a server file system for reading only
      - Returns a reference to the named file system, which must be a server file system. This file system can be used for reading only. It is not possible to write into it.
        Currently the only supported file system is the server's configuration („config-runtime“). That way it is possible to include additional resources like config files or XSLT stylesheets, which are not stored as resource assets.
    - Attributes
      - „name“: „com.censhare.api.io.GetFilesystemRef“
      - „returning“: The URL identifying the resulting directory or archive file. It can be used as a „censhare:result-file-locator“ return parameter
    - Parameters
      - „name“: The name of the file system

XSLT Commands

censhare

- I/O
  - Get file system reference
    - Example
      - <!-- get file system reference of config runtime -->
        <cs:command name="com.censhare.api.io.GetFilesystemRef" returning="config"/>
        <cs:param name="name" select="'config-runtime'"/>
        </cs:command>
        <!-- execute transformation with stylesheet of config runtime file system -->
        <cs:command name="com.censhare.api.transformation.XslTransformation" returning="xml">
        <cs:param name="stylesheet" select="concat($config, 'modules/content_export/stylesheets/export.xsl'"/>
        <cs:param name="source" select="asset/storage_item[@key='master'][1]"/>
        </cs:command>

XSLT Commands

censhare

- Renderer
  - Commands
    - Render

XSLT Commands

- Renderer
  - Render
    - Executes the rendering commands given through the Render-Client
    - Same syntax of XML message exchanging between censhare Server and Render-Client
    - Attributes
      - „name“: „com.censhare.api.Renderer.Render“
      - „returning“: The URL identifying the resulting directory or archive file. It can be used as a „censhare:result-file-locator“ return parameter
    - Parameters
      - „facility“: String, currently allowed values: „indesign-6.0“ for CS-4, „indesign-7.0“ for CS-5, „indesign-7.5“ for CS-5.5, „indesign-8.0“ for CS-6
      - „timeout“: timeout in seconds (default is 120 seconds)

XSLT Commands

censhare

- Renderer
  - Render
    - Methods
      - check-missing-fonts, check-placed-collections, close, copy-pagearea, copy-pagearea-asset, correct-document, debug, detach, detach-pagearea, edit-placement, eps, mark-not-printable-boxes, move-box, move-pagearea, open, pdf, place, place-collection, preview, rename-group, save, script, text-content, update-asset-element-structure, update-auto-client-variant, update-content, update-geometry, update-placeholder,

XSLT Commands

censhare

- Renderer
  - Render
    - Example
      - <cs:command name="com.censhare.api.Renderer.Render">
        <cs:param name="facility" select="{concat('indesign-', $indesignStorageItem/@app_version)}"/>
        <cs:param name="renderer-transformation">
        <cmd>
        <renderer>
        <command method="open" asset_id="{asset[1]/@id}" document_ref_id="1"/>
        <command method="preview" spread="false" asset_element_id="1" document_ref_id="1"/>
        <command method="pdf" spread="false" document_ref_id="1"/>
        <command method="close" document_ref_id="1"/>
        </renderer>
        <assets>
        <xsl:copy-of select="$asset"/>
        </assets>
        </cmd>
        </cs:param>
        </cs:command>

More info

censhare

- Links
  - W3C – XSLT 2.0
    - http://www.w3.org/TR/xslt20/
  - Wikipedia – XSLT
    - http://en.wikipedia.org/wiki/Xslt
  - Zvon.org – XSLT Tutorial in English and German
    - http://zvon.org/xxl/XSLTutorial/Output/index.html
    - http://zvon.org/xxl/XSLTutorial/Output_ger/index.html

End – Thank you