IBM CONTENT MANAGER ONDEMAND WEB ENABLEMENT KIT JAVA API

GENERIC TRANSFORM INTERFACE

April 29, 2019

Author: Bob Lichens

Revisions: Sandi Pond

IBM Content Management
OnDemand Development Team

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This article will walk through the basics of how to implement a new interface for 3rd party applications to be integrated into the IBM® Content Manager OnDemand Web Enablement Kit Application Programming Interface set (referred to hereafter as simply ODWEK). This interface will be used in conjunction with the document retrieval functions.

What is IBM Content Manager OnDemand Web Enablement Kit Java API (ODWEK)?

The ODWEK Java API provides industry standard Java™ classes that can be used by a customer to write a custom web application that can access data stored on the Content Manager OnDemand server. This custom application can, for example, permit the end user to logon to a Content Manager OnDemand (CMOD) server, get a list of folders, search a specific folder, generate a hit list of matching documents, and retrieve those documents for viewing. The API also contains many functions that provide advanced functionality.

ODWEK Java API Coding "Best Practices"

The Content Manager OnDemand development team has prepared a document with recommendations of how to best use the ODWEK Java API. Please read the IBM Redbook "IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond" http://www.redbooks.ibm.com/abstracts/sg247646.html to make sure you fully understand how the ODWEK Java API interfaces with the JVM and Content Manager OnDemand Systems to avoid common coding mistakes.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

OVERVIEW

Prior to Version 8.5.0.0, the ODWEK Java API provided a tight integration with only a few specific transforms. These include AFP2PDF, AFP2HTML, AFP2XML, and Xenos transforms. These transform engines can be used by the ODWEK API to generate different document types for display purposes. While this provided invaluable functionality to our customers, it meant that new transform engines could not be readily integrated into ODWEK.

To meet this requirement, we have added a highly flexible interface, known as the Generic Transform Interface, to the ODWEK Java API that allows a developer to implement most any document processing solution they desire.

This new ODWEK Interface allows a developer to implement an external program to transform a document in one of two ways.

1. Basic Implementation. If the transform vendor provides a basic command line executable, this is the easiest way to implement a transform within ODWEK and will be supported in an XML interface to include hooks to not only get all of the document detail stored in CMOD, but also allow the pass through of transform specific options.
2. Advanced Implementation. The ODWEK Java API also provides a Java interface that a developer can use to add even more flexibility to their client solution. The Java interface allows a developer to get the document byte stream from ODWEK, then use any method(s) they desire to convert the document. This could include calls to webservices that would allow remote transformation. Once the document is transformed, the resulting data can be returned to ODWEK, where it will be passed back to the caller.

Installation Note: To enable the Generic Transform Interface in ODWEK, an XML document must be created and defined in the ODConfig.Properties object. This is identified with the <ODConfig.TransformXML> keyname and must include the fully qualified path to the XML file where the transforms have been defined.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Basic Implementation: Executable Interface

The basic implementation of the Generic Transform interface involves an XML configuration file that defines a transform to ODWEK. ODWEK uses the values in the XML file to utilize the transform's command line (cmdline) executable functionality. With this configuration you can request details that CMOD has stored for the document to be passed in as cmdline options and also pass through transform specific options as identified in the transform.xml.

Figure 1 Sample XML for Basic Implementation below shows an XML file that defines a customer transform that could be used in this implementation.

```
<Transforms>
   <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <ODFileType>L</ODFileType>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdParms>
        <RECORDLENGTH>-lm</RECORDLENGTH>
        <CARRIAGECONTROL>-x</CARRIAGECONTROL>
        <CODEPAGE>-a</CODEPAGE>
        <OUTPUTFILE>-o</OUTPUTFILE>
    </CmdParms>
    <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
    <Passthru>
        <!-- Use tag cmdlineparm to declare additional cmdline variables -->
        <!-- that the transform might require -->
        <Cmdlineparm>-r PDF</Cmdlineparm>
    </Passthru>
  </transform>
<Transforms>
```

Figure 1 Sample XML for Basic Implementation

In this example, a transform has been defined named "MyTXFRM_EXE", which will call the transform command "txfrm.exe" defined by the <CmdLineExe> tag. The value given for the <TransformName> tag, in this example MyTXFRM_EXE, is specified as the "viewer" value argument in the ODWEK retrieve call. For example,

```
byte [] transformed_data_bytes = odHit.retrieve("MyTXFRM_EXE");
```

Additionally, from this XML, ODWEK will know that the transform requires RECORDLENGTH, CARRIAGECONTROL, CODEPAGE, and OUTPUTFILE information from OnDemand. The txfrm.exe requires some

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

additional information to be passed along on the cmdline as well. The "-r" specified in <Cmdlineparm> tag has no meaning to CMOD, so it is just passed through "as is" and set on the cmdline call to the txfrm.exe.

If the hit being retrieved/transformed has the characteristics shown in *Figure 2 Application View Information*

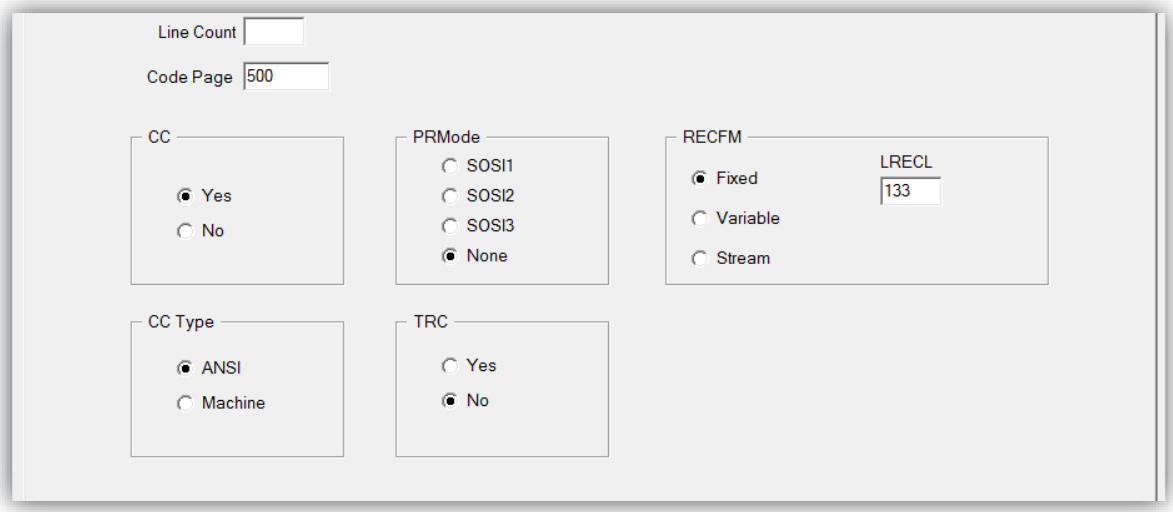

Figure 2 Application View Information

and using the example XML shown above, the ODWEK Generic Transform Interface takes the retrieved bytes, writes them to a temporary file <datafilename> to be used as an argument to txfrm.exe, and generates and executes the following command:

c:/opt/txfrm.exe -lm 133 –x A –a 500 –o <outputfilename> -r PDF <datafilename>

The txfrm.exe writes its output to <outputfilename>. ODWEK reads the output and returns it to the caller as a byte array. The byte array named "transformed_data_bytes" will then contain the transformed data.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

V9.5 Enhancement: Customizing values returned from ODWEK

For some transforms, values returned from ODWEK may not be consistent with the command line values expected by the transform. For example, a transform may have a fixed set of options to specify a carriage control type. The values returned by ODWEK when the <CARRIAGECONTROL> tag is included in the <CmdParms> are 'A' (ANSI), 'M' (Machine), and 'N' (None). Recall that the command produced by the XML in *Figure 1 Sample XML for Basic Implementation* is

c:/opt/txfrm.exe -lm 133 –x A –a 500 –o <outputfilename> -r PDF <datafilename>

Because the <CARRIAGECONTROL> tag is present, ODWEK will return the document's corresponding value - "-x A", or "-x M", or "-x N" depending on the CC Type defined in this document's Application definition. If the transform defines a different set of acceptable values – for example 2, 4, and 0 - to specify the document's carriage control, mapping those values can be accomplished by substituting the following XML:

```
<Transforms>
   <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <ODFileType>L</ODFileType>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdParms>
        <RECORDLENGTH>-lm</RECORDLENGTH>
        <CCANSI>-x 2</CCANSI>
        <CCMACHINE>-x 4</CCMACHINE>
        <CCNONE>-x 0</CCNONE>
        <CODEPAGE>-a</CODEPAGE>
        <OUTPUTFILE>-o</OUTPUTFILE>
    </CmdParms>
    <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
    <Passthru>
     <!-- Use tag cmdlineparm to declare additional cmdline variables that the transform might require
        <Cmdlineparm>-r PDF</Cmdlineparm>
    </Passthru>
  </transform>
<Transforms>
```

Figure 3 Sample XML with Custom Options

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Note that the <CARRIAGECONTROL> node has been replaced by three values. When the CC Type returned by ODWEK matches ANSI, rather than an 'A', the command will include "-x 2".

This type of substitution can be used to specify the RECFM (Record Format), PRMode, TRC, as well as CC Type. See Appendix B for valid node names.

V9.5 Enhancement: Application Group/Application specific XML

In version 9.5.0.2, ODWEK now provides additional options under the <transform> node that allow the transform command parameters to be generated based on an Application Group or Application Group and Application pair.

Figure 4 Sample XML with <ApplicationGroup><Application> tags below shows a sample transform.xml that could be used in this implementation.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

```
<Transforms>
   <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <ODFileType>L</ODFileType>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
    <CmdParms>
        <RECORDLENGTH>-lm</RECORDLENGTH>
        <CARRIAGECONTROL>-x</CARRIAGECONTROL>
        <CODEPAGE>-a</CODEPAGE>
        <OUTPUTFILE>-o</OUTPUTFILE>
    </CmdParms>
    <Passthru>
     <!-- Use tag cmdlineparm to declare additional cmdline variables -->
     <!-- that the transform might require -->
        <Cmdlineparm>-r PDF</Cmdlineparm>
    </Passthru>
    <ApplicationGroup name='FinancialReports'>
        <CmdParms>
            <CARRIAGECONTROL>-x</CARRIAGECONTROL>
            <CODEPAGE>-c</CODEPAGE>
            <OUTPUTFILE>-o</OUTPUTFILE>
        </CmdParms>
        <Passthru>
            <Cmdlineparm>-h 612 -w 1008</Cmdlineparm>
        </Passthru>
    </ApplicationGroup>

    <ApplicationGroup name='SalesReports'>
        <CmdParms>
            <CARRIAGECONTROL>-x</CARRIAGECONTROL>
            <CODEPAGE>-c</CODEPAGE>
            <OUTPUTFILE>-o</OUTPUTFILE>
        </CmdParms>

        <Application name='Forecasts'>
            <CmdParms>
                <CARRIAGECONTROL>-x</CARRIAGECONTROL>
                <CODEPAGE>-c</CODEPAGE>
                <OUTPUTFILE>-o</OUTPUTFILE>
                <TRC_EXIST>-t</TRC_EXIST>
            </CmdParms>
            <Passthru>
                <Cmdlineparm>-h 1008 -w 612</Cmdlineparm>
            </Passthru>
        </Application>
    </ApplicationGroup>
  </transform>
<Transforms>
```

Figure 4 Sample XML with <ApplicationGroup><Application> tags

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

The following table shows the transform command that will be generated based on the sample XML and Application Group and Application of the document being retrieved:

Doc's AG	Doc's Appl	XML Match for AG	XML Match for Appl	Command generated
FinancialReports	Ledgers	Yes	No	c:/opt/txfrm.exe –x A –c 850 –o <outputfilename> -h 612 –w 1008 <datafilename>
SalesReports	Forecasts	Yes	Yes	c:/opt/txfrm.exe –x N –c 500 –t 1 –o <outputfilename> -h 1008 –w 612 <datafilename>
SalesReports	WeeklySummaries	Yes	No	c:/opt/txfrm.exe –x A –c 500 –o <outputfilename> <datafilename>
Accounting	Payable	No	No	c:/opt/txfrm.exe -lm 133 –x M –a 500 –o <outputfilename> -r PDF <datafilename>

Figure 5 Table of Generated Commands

Note: Inheritance is not supported. If an <ApplicationGroup> node is matched then only those options within that node will be used for the transform, no parameters found for a parent transform node will be used. Similarly, if an <Application> node is matched within an <ApplicationGroup> node, only those options will be used for the transform, nothing from the <ApplicationGroup> node will be used.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Advanced Implementation: Custom Java Interface

The Advanced Implementation of the Generic Transform Interface allows developers to write a Java interface to ODWEK that can handle the transform requests in a programmatic way, allowing for the most application flexibility. Developers can create a class and implement the transformData() method to accept document data and details from CMOD and transform the data in any way they see fit.

Figure 6 Sample XML for Advanced Implementation below shows a sample of the transform.xml that could be used in this implementation.

```
<Transforms>
    <transform>
    <TransformName>MYTXFRM</TransformName>
    <TransformDescription>GENERIC Transform Engine</TransformDescription>
    <ClientClass>com.companyA.corp.CustomTransform</ClientClass>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdParms>
      <AG_NAME>agName</AG_NAME>
      <APPL_NAME>applName</APPL_NAME>
      <RECORDFORMAT>recfmt</RECORDFORMAT>
      <RECORDLENGTH>LineLength</RECORDLENGTH>
      <CARRIAGECONTROL>CC</CARRIAGECONTROL>
      <CODEPAGE>CodePage</CODEPAGE>
    </CmdParms>
    </transform>
</Transforms>
```

Figure 6 Sample XML for Advanced Implementation

As with the basic implementation, this XML stanza allows the developer to set up which details are required for document transformation and how those details should be passed to the Java transform interface. See *Figure 7 Sample Code for Advanced Implementation* for an example of how the <ClientClass> can be used with the XML stanza above to access the document data.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

```
//*************************************************************
// Testcase:  CustomTransform
//
// This class tests the ODWEK Generic Transform's Custom
// Java Interface by implementing the required transformData method.
//
// transformData is called by ODWEK when its corresponding custom
// viewer is called via ODHit.retrieve.
//*************************************************************
import java.util.*;
import com.ibm.edms.od.*;

public class  CustomTransform {
    public static HashMap transformData(HashMap odMap) throws Exception {

        // List this transform name from the XML file
        System.out.println("  Transform name: " + (String)odMap.get(ODTransform.TXFRM_REQ_NAME));

        // List the property keys and values ODWEK read from the transform XML
        // file and provided to this Custom Class
        System.out.println("  Transform properties:");
        Properties gtProps = (Properties)odMap.get(ODTransform.TXFRM_REQ_PROPS);
        Enumeration<?> enumeration = gtProps.keys();
        List<String> list = new ArrayList<String>();
        while (enumeration.hasMoreElements()) {
            list.add((String)enumeration.nextElement());
        }
        Collections.sort(list);
        for (String key : list)
            System.out.println(String.format("%25s = %-25s", key, gtProps.getProperty(key)));

        // Retrieve the native document from ODWEK
        byte[] inDoc = (byte [])odMap.get(ODTransform.TXFRM_REQ_DATA);
        System.out.println("  Native document size: " + (inDoc == null ? null: inDoc.length));

        // Retrieve the document resources from ODWEK
        byte[] inRes = (byte [])odMap.get(ODTransform.TXFRM_REQ_RES);
        System.out.println("  Native doc resource size: " + (inRes == null ? null: inRes.length));

        // Normally this is where you do the transform or do something with the byte data.
        // Let's just concat the resources if there are any to the doc
        byte[] transformedDoc;
        if (inRes != null) {
            transformedDoc  = new byte[inRes.length + inDoc.length];
            System.arraycopy(inRes, 0, transformedDoc, 0, inRes.length);
            System.arraycopy(inDoc, 0, transformedDoc, inRes.length, inDoc.length);
        }
        else
            transformedDoc = inDoc;
        System.out.println("  Concatenated resources to doc size: " + transformedDoc.length);

        // Send the transformed data back to ODWEK
        HashMap rtnMap = new HashMap();
        rtnMap.put(ODTransform.TXFRM_RESP_DATA, transformedDoc);
        return rtnMap;
    }
}
```

Figure 7 Sample Code for Advanced Implementation

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

The transform class in *Figure 7 Sample Code for Advanced Implementation* is utilized by calling the document retrieval with the <TransformName> as the viewer parameter:

```
byte [] transformed_data_bytes = odHit.retrieve("MYTXFRM");
```

As you can see in the example, by setting up the HashMap to pass document byte arrays in and out of this custom interface, as well as defining a custom Java class that contains the "transformData" method, ODWEK can provide developers the flexibility to get the raw document data from ODWEK, gather all the document details that CMOD might have stored from the loading of the data, and then transform that document. The transformed document data can be passed back through ODWEK to the original request.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Appendix A: XmlTag Names for Transform Specification

Xml Tag Name	ODTransform.FIELD	Info
TransformName	TransformName	Name of the Transform. This will be used as the Viewer argument passed to ODWEK Retrieve function.
TransformDescription	TRANSFORM_DESC	Description of transform.
ClientClass	TRANSFORM_CLIENTCLASS	The classname of the custom interface class.
CmdLineExe	TRANSFORM_CMDLINEEXE	Fully qualified name of the transform executable.
ODFileType	TRANSFORM_INPUTFILETYPE	The input file type handled by the transform.
OutputMimeType	TRANSFORM_MIMETYPE	The mimetype of the data as it is returned from the transform.
OutputExtension	TRANSFORM_OUTPUTEXT	The extension of the data that is returned from the transform.
CmdParms	TRANSFORM_PARMS	These are the mappings of OD Values to custom variables. See Constant key words below.
Passthru	TRANSFORM_PASSTHRU	These values are passed through ODWEK directly to the transform.
Cmdlineparm	TRANSFORM_PASSTHRU_CMDLINE	These values are passed through ODWEK directly to the transform cmdline.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Appendix B: Document details XMLTags

Xml Tag Name	ODTransform.FIELD	Info
RECORDFORMAT	DOCUMENT_RECORD_FORMAT	The Record Format of the document as stored in CMOD.
RECORDLENGTH	DOCUMENT_RECORD_LENGTH	The Record Length of the document as stored in CMOD.
CARRIAGECONTROL	DOCUMENT_CARRIAGE_CONTROL	The Carriage Control of the document as stored in CMOD.
TRC_EXIST	DOCUMENT_TRC_EXIST	The TRC settings as stored in CMOD.
DOCROTATION	DOCUMENT_ROTATION	The rotation of the document as stored in CMOD.
AG_NAME	AGNAME	The CMOD Application Group where the document is stored.
APPL_NAME	APPLNAME	The CMOD Application where the document is stored.
CODEPAGE	DOCUMENT_CODEPAGE	The codepage of the document as stored in CMOD.
LINEDELIMITER	DOCUMENT_LINE_DELIMITER	The line delimiter of the document as stored in CMOD.
INPUTFILE	TXFRM_INPUT_FILE	The Inputfile parameter to be used by the transform.
OUTPUTFILE	TXFRM_OUTPUT_FILE	The OutputFile parameter to be used by the transform.
V9.5 ENHANCEMENTS		
CCANSI	DOCUMENT_CC_ANSI	Used instead of <CARRIAGECONTROL> to define the command line option and value when the document's CC Type is 'ANSI' as stored in CMOD.
CCMACHINE	DOCUMENT_CC_MACHINE	Used instead of <CARRIAGECONTROL> to define the command line option and value when the document's CC Type is 'Machine' as stored in CMOD.
CCNONE	DOCUMENT_CC_NONE	Used instead of <CARRIAGECONTROL> to define the command line option and value when the document's CC is 'No' as stored in CMOD.
RECORDFORMATFIXED	DOCUMENT_RECORDFORMAT_FIXED	Used instead of <RECORDFORMAT> to define the command line option and value when the document's

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

		RECFM is 'Fixed' as stored in CMOD.
RECORDFORMATVARIABLE	DOCUMENT_RECORDFORMAT_VARIABLE	Used instead of <RECORDFORMAT> to define the command line option and value when the document's RECFM is 'Variable' as stored in CMOD.
RECORDFORMATSTREAM	DOCUMENT_RECORDFORMAT_STREAM	Used instead of <RECORDFORMAT> to define the command line option and value when the document's RECFM is 'Stream' as stored in CMOD.
PRMODENONE	DOCUMENT_PRMODENONE	Used instead of <PRMODE> to define the command line option and value when the document's PRMode is 'None' as stored in CMOD.
PRMODESOSI1	DOCUMENT_PRMODESOSI1	Used instead of <PRMODE> to define the command line option and value when the document's PRMode is 'SOSI1' as stored in CMOD.
PRMODESOSI2	DOCUMENT_PRMODESOSI2	Used instead of <PRMODE> to define the command line option and value when the document's PRMode is 'SOSI2' as stored in CMOD.
TRCYES	DOCUMENT_TRCYES	Used instead of <TRC_EXISTS> to define the command line option and value when the document's TRC is 'Yes' as stored in CMOD.
TRCNO	DOCUMENT_TRCNO	Used instead of <TRC_EXISTS> to define the command line option and value when the document's TRC is 'No' as stored in CMOD.

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Appendix C: ODWEK Client HashMap Keys (Advanced Java Implementation)

TXFRM_RESP_DATA	This is the Hashmap key for the transformed data byte[] to be returned to ODWEK.
TXFRM_REQ_NAME	Name of Transform for this Request.
TXFRM_REQ_METHOD	Method Name used in the custom Java class. A Method "transformData" must exist in the client class.
TXFRM_REQ_DATA	Original Content Manager OnDemand document data contained in this request.
TXFRM_REQ_PROPS	Document details as specified/requested in the transfrom.xml.
TXFRM_REQ_RES	Content Manager OnDemand Resource data contained in the request (where applicable).

© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Appendix D: Defining the ODFileType in the transform config XML

In order to use the method ODServer.getTransformViewersForDocType(char doctype), an <ODFileType> node must be included as a child node of the <transform> node in the XML file that defines the transforms. Use the following values to represent the proper file type.

Data Type	ODFileType Character
AFP	A
BMP	B
SCS Extended	C
Global DJDE	D
JFIF (JPEG)	F
GIF	G
LINE	L
METACODE	M
PDF	P
PNG	Q
SCS	S
TIFF	T
User Defined	U
PCX	X
XML	Z

For example,

```
<transform>
        <TransformName>OD_AFP2PDF</TransformName>
        <ODFileType>A</ODFileType>
…
</transform>
```


© **Copyright 2017-2019 All Rights Reserved, UNICOM Systems, Inc. – a division of UNICOM Global.**.

© **Copyright International Business Machines Corporation 2015, 2019.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.