

# RSS Ingestor for XAM Reference VIM

**Finnbarr P. Murphy**

(fpm@fpmurphy.com)

Recently the [SNIA XAM SDK TWG](#) (Software Development Kit Technical Working Group) released a new version (version 0.7) of the [XAM SDK](#) with lots of new features. Included in this SDK was the 5th code drop for the Reference VIM. Here is a summary of the major new functionality included in this code drop.

- Support for XSet hold and release
- Support for XSet autodelete and shred policies
- Support for simple where clauses in level 1 queries
- Support for retention policies including XSet import and export
- Support for asynchronous methods
- Support for level 2 queries

Also recently released were [version 1.01](#) of the XAM technical positions (AKA specifications). If you are familiar with version 1.0 of these three documents you may just want to read the errata for each document which is also available on the website.

I decided to test the Reference VIM by writing a small Java application to ingest the RSS feed from my blog, display some information about each post and store this information in the Reference VIM database. This application uses an XML configuration file to store a list of the RSS feeds to be ingested and the popular [ROME](#) Java library to ingest and parse the individual RSS feeds. Rather than re-invent the wheel, it also uses a modified version of the Java source file *ExampleBase.java* which is supplied as part of the latest XAM SDK to create and authenticate the connection to the XAM library and Reference VIM.

Here is *RSS2XAM.java* which is meat of the application. As you can see, *RSS2XAM* extends *ExampleBase*. This is a useful paradigm for writing applications which use the Reference VIM.

```
//  
// RSS2XAM - Ingest RSS feed and add to XAM Storage System  
// using SNIA XAM SDK V0.7 reference VIM  
//  
  
import java.net.URL;  
import java.net.URLConnection;  
import java.util.List;  
import java.util.Calendar;  
import java.io.File;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import org.w3c.dom.*;  
  
import com.sun.syndication.feed.synd.SyndFeed;  
import com.sun.syndication.feed.synd.SyndEntry;  
import com.sun.syndication.feed.synd.SyndImageImpl;  
import com.sun.syndication.io.SyndFeedInput;  
import com.sun.syndication.io.XmlReader;  
import com.sun.syndication.feed.synd.SyndContentImpl;
```

```

import com.sun.syndication.feed.synd.SyndCategory;

import org.snia.xam.XSet;
import org.snia.xam.XSystem;
import org.snia.xam.XUID;

public class RSS2XAM
    extends ExampleBase
{
    private static void iterateRSSFeed(String rssFeedUrl, XSystem xsystem)
        throws Exception {

        URLConnection feedUrl = new URL(rssFeedUrl).openConnection();
        SyndFeedInput input = new SyndFeedInput();
        SyndFeed feed = input.build(new XmlReader(feedUrl));

        System.out.println("RSS feed: " + rssFeedUrl + " (" + feed.getFeedType() + ")");

        // ingest the feed, display info and create the XSets
        List list = feed.getEntries();
        for (int i = 0 ; i < list.size(); i++)
        {
            SyndEntry entry = (SyndEntry)list.get(i);

            String display = "Entry: " + i;
            display += "\n Title: " + entry.getTitle();
            display += "\n Link: " + entry.getLink();
            display += "\n Author: " + entry.getAuthor();
            display += "\n Date Published: " + entry.getPublishedDate();

            // get the list of categories
            List catList = entry.getCategories();
            String categories = "";
            for (int j = 0 ; j < catList.size(); j++)
            {
                if (j > 0 ) categories += ", ";
                SyndCategory cat = (SyndCategory)catList.get(j);
                categories += cat.getName();
            }

            display += "\n Categories: " + categories;
            System.out.println(display);

            XSet xset = xsystem.createXSet(XSet.MODE_UNRESTRICTED);
            xset.createProperty("com.fpmurphy.rss.title", false, entry.getTitle());
            xset.createProperty("com.fpmurphy.rss.link", false, entry.getLink());
            xset.createProperty("com.fpmurphy.rss.author", false, entry.getAuthor());
            xset.createProperty("com.fpmurphy.rss.categories", false, categories);
            Calendar cal=Calendar.getInstance();
            cal.setTime(entry.getPublishedDate());
            xset.createProperty("com.fpmurphy.rss.published_date", false, cal);

            XUID xuid = xset.commit();
            System.out.println(" XSet XUID ==> " + xuid.toString() + "\n");
            xset.close();
        }
    }

    public static void main(String argv[]) {

        try {
            File file = new File("rss.xml");

            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document document = db.parse(file);
            document.getDocumentElement().normalize();
        }
    }
}

```

```

// init, load Reference VIM, and authenticate
System.out.println("Connecting to Reference VIM");
initLibrary();
XSystem xsystem = connectToVIM(s_xri);
authenticate(xsystem);

// parse the rss.xml
NodeList nodeList = document.getElementsByTagName("feed");
int totalFeed = nodeList.getLength();
System.out.println("Number of Feeds: " + totalFeed);

for (int i = 0; i < nodeList.getLength(); i++) {
Node feedNode = nodeList.item(i);
if (feedNode instanceof Element) {
Element child = (Element) feedNode;
String check = child.getAttribute("check");
String rssFeedUrl = child.getTextContent();

// hand off to do the work
iterateRSSFeed(rssFeedUrl, xsystem);
}
} catch (Exception e) {
e.printStackTrace();
}
}
}
}
}

```

Here is a copy of the modified *exampleBase.java*. If you compare it to the original source code in the XAM SDK (*.../Java\_Reference\_VIM/examples/ExampleBase.java*) you will see that most of the changes are minor simplifications.

```

/*
 * Copyright (c) 2009, Sun Microsystems, Inc.
 * Copyright (c) 2009, The Storage Networking Industry Association.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 *
 * Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 *
 * Neither the name of The Storage Networking Industry Association (SNIA) nor
 * the names of its contributors may be used to endorse or promote products
 * derived from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
 * THE POSSIBILITY OF SUCH DAMAGE.
 */

import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;

```

```

import java.util.Properties;
import org.snia.xam.XAMException;
import org.snia.xam.XAMLibrary;
import org.snia.xam.XSystem;
import org.snia.xam.base.XAMImplementation;
import org.snia.xam.util.SASLUtills;
import org.snia.xam.vim.reference.ReferenceAuthenticationStatus;
import org.snia.xam.vim.reference.utils.ReferenceSaslUtills;

public class ExampleBase {

    public static final String PROP_FILE = "rss2xam.props";
    public static final String XRI_PROP = "xam.xri";
    public static final String CONFIG_PROP = "xam.vims";
    public static final String USER_PROP = "xam.username";
    public static final String PASS_PROP = "xam.password";

    protected static final String DEFAULT_USER = "test";
    protected static final String DEFAULT_PASS = "test";
    protected static String s_pass;
    protected static String s_user;
    protected static XAMLibrary s_xam;
    protected static String s_xri;

    public static void initLibrary() throws Exception {
        System.out.println("\nInitializing VIM");

        Properties props = new Properties();
        String testPropFile = System.getProperty(PROP_FILE);
        if (testPropFile == null)
            testPropFile = PROP_FILE;

        System.out.println("Loading properties from file: " + testPropFile);
        props.load(new FileInputStream(testPropFile));

        s_xri = props.getProperty(XRI_PROP);

        s_user = props.getProperty(USER_PROP, DEFAULT_USER);
        s_pass = props.getProperty(PASS_PROP, DEFAULT_PASS);

        System.out.println("VIM Configuration contained in file: " + props.getProperty(CONFIG_PROP));
        s_xam = new XAMImplementation(props.getProperty(CONFIG_PROP));
    }

    public static XSystem connectToVIM(String xri) throws Exception {
        XSystem xsystem = null;
        System.out.println("Connection arguments: " + xri);
        xsystem = s_xam.connect(xri);
        return xsystem;
    }

    public static void authenticate(XSystem system) throws XAMException {
        String defMech = system.getString(XSystem.XAM_XSYSTEM_AUTH_SASL_DEFAULT);
        ByteArrayOutputStream response = new ByteArrayOutputStream(200);
        byte[] inputData = null;
        int retValue = 0;
        if (defMech.equals(ReferenceSaslUtills.SASL_MECHANISM_ANONYMOUS)) {
            retValue = system.authenticate(inputData, response);
        } else if (defMech.equals(SASLUtills.SASL_PLAIN)) {
            byte[] creds = ReferenceSaslUtills.encodeSASLPlain(null,
                ReferenceAuthenticationStatus.TEST_USERNAME,
                ReferenceAuthenticationStatus.TEST_PASSWORD);
            retValue = system.authenticate(creds, response);
        } else {
            throw new XAMException("Unknown SASL mechanism " + defMech);
        }
    }
}

```

```

        if (retValue != XSystem.XAM_SASL_COMPLETE) {
            throw new XAMException("Failed to authenticate.");
        }
    }
}

```

Here is a copy of my *rss2xam.props*. I am storing the generated XSets at */home/fpm/rss/store* but you can specify any directory you like by modifying the *dir=* directive on the second line of this file. Make sure that this directory exists and is writeable or you will get an exception when you invoke the application. I will leave it to you to modify the source to test that the specified directory exists and is writeable. As you can see, this file also specifies that the Reference VIM configuration file (*vim.config*) exists in the current directory. Again you are free to change the location to wherever you like.

```

xam.vims=./vim.config
xam.xri=snia-xam://SNIA_Reference_VIM!localhost?dir=/home/fpm/rss/store
xam.username=testuser
xam.password=testpasswd

```

Here is my *vim.config*: It is basically the same as comes with the XAM SDK. The only difference is that I create a new log every time the application is invoked rather than appending to the existing log.

```

.xam.config.vim.alias.SNIA_Reference_VIM=org.snia.xam.vim.reference.ReferenceVIM
xam_boolean..xam.log.append=false
xam_int..xam.log.max.size=200
xam_int..xam.log.verbosity=1
xam_int..xam.log.level=5

```

Here is a copy of my *rss.xml*. It contains a simple feed i.e. this blog. You can add more feeds like you like. The *check* attribute is for future use and is not used by this application.

```

http://blog.fpmurphy.com/feed

```

Here is the output when this application is compiled and invoked.

```

$ javac RSS2XAM.java ExampleBase.java
$ java RSS2XAM
Connecting to Reference VIM
Initializing VIM
Loading properties from file: rss.props
VIM Configuration contained in file: ./vim.config
Connection arguments: snia-xam://SNIA_Reference_VIM!localhost?dir=/home/fpm/rss/store

Number of Feeds: 1
RSS feed: http://blog.fpmurphy.com/feed (rss_2.0)

Entry: 0
  Title: Amazon SimpleDB Typica Tutorial
  Link: http://blog.fpmurphy.com/2009/08/amazon-simpledb-typica-tutorial.html
  Author: fpmurphy
  Date Published: Tue Aug 04 12:35:14 GMT-05:00 2009
  Categories: AWS, Java, XAM
  XSet XUID ==> AAA6AwAqbu8xMjUwNDc20TYwNzcwAcSLfePaTUlTnoaEkQ5kB0gDBBh2

```

Entry: 1

Title: XAM Query Language  
Link: <http://blog.fpmurphy.com/2009/07/xam-query-language.html>  
Author: fpmurphy  
Date Published: Sun Jul 19 10:10:41 GMT-05:00 2009  
Categories: XAM, XSet, XStream, XSystem, SNIA  
XSet XUID ==> AAA6AwAqbH8xMjUwNDc20TYw0DExAjloAZmNdUUPto94Weh/11seuv3/

Entry: 2

Title: Linux HPET Support  
Link: <http://blog.fpmurphy.com/2009/07/linux-hpet-support.html>  
Author: fpmurphy  
Date Published: Mon Jul 06 02:50:03 GMT-05:00 2009  
Categories: C, HPET, Linux  
XSet XUID ==> AAA6AwAqJ0xMjUwNDc20TYw0DMxAxk6BG70QErMh3iqx3BMNwYGbeYJ

Entry: 3

Title: XAM Mandated Fields  
Link: <http://blog.fpmurphy.com/2009/06/xam-mandated-fields.html>  
Author: fpmurphy  
Date Published: Mon Jun 29 21:07:04 GMT-05:00 2009  
Categories: XAM, SNIA  
XSet XUID ==> AAA6AwAqUWoxMjUwNDc20TYw0DUxBM5rT72XVEEeuQvgMU1G47JugiVN

Entry: 4

Title: Atahualpa Theme  
Link: <http://blog.fpmurphy.com/2009/06/blog-now-uses-wordpress-with-atahualpa-theme.html>  
Author: fpmurphy  
Date Published: Wed Jun 24 16:14:55 GMT-05:00 2009  
Categories: Uncategorized  
XSet XUID ==> AAA6AwAqGhAxMjUwNDc20TYw0DY5BbZxkkmfZUMmnKAsE/THFApESwu7

Entry: 5

Title: XAM Canonical Format  
Link: <http://blog.fpmurphy.com/2009/06/xam-and-xop.html>  
Author: fpmurphy  
Date Published: Sat Jun 20 17:15:00 GMT-05:00 2009  
Categories: XAM, XOP, SNIA  
XSet XUID ==> AAA6AwAq2AkxMjUwNDc20TYw0Dg4Bhk10Dii0U0QlaAbfG+dvMwBUxFk

Entry: 6

Title: Fedora 11 New Extended File Attributes Namespace  
Link: <http://blog.fpmurphy.com/2009/06/fedore-11-extended-attibutes-namespace.html>  
Author: fpmurphy  
Date Published: Mon Jun 15 09:31:00 GMT-05:00 2009  
Categories: Extended Attributes, Fedora 11, XAM, Linux  
XSet XUID ==> AAA6AwAqa8QxMjUwNDc20TYw0TA3B3Mj+txXN0YJjufntzTjYw5uPlqR

Entry: 7

Title: Fedora 11 nVidia Twinview Support  
Link: <http://blog.fpmurphy.com/2009/06/fedora-11-nvidia-twinview-support.html>  
Author: fpmurphy  
Date Published: Thu Jun 11 12:07:00 GMT-05:00 2009  
Categories: Fedora 11, Linux, Twinview, nVidia, Fedora  
XSet XUID ==> AAA6AwAq4JQxMjUwNDc20TYw0TI5CNTTrNQ44okBPnnJFleb57UFNj5t1

Entry: 8

Title: Linux Security Capabilities  
Link: <http://blog.fpmurphy.com/2009/05/linux-security-capabilities.html>  
Author: fpmurphy  
Date Published: Thu May 28 10:14:00 GMT-05:00 2009  
Categories: Fedora 11, Linux, POSIX.1e, linux capabilities  
XSet XUID ==> AAA6AwAqURQxMjUwNDc20TYw0TU1CTY+vTycKkMhgRj22XFJ+CpMdnKG

Entry: 9

Title: Windows Parallel Filesystems  
Link: <http://blog.fpmurphy.com/2009/05/windows-parallel-filesystems.html>  
Author: fpmurphy

```
Date Published: Tue May 26 12:27:00 GMT-05:00 2009
Categories: Windows parallel filesystem, Windows
XSet XUID ==> AAA6AwAq/jsxMjUwNDc20TYw0Tg0CuIRPc1YkUozqtuI+oCaeTkQx4n7
```

```
$
```

As you can see from the following listing, the Reference VIM creates an XML file and a subdirectory for each XSet that it creates as well as a Derby database called *ReferenceVimDB*. The name of each XML file and corresponding subdirectory is the XUID of the XSet.

```
$ ls store
ReferenceVimDB                                     XSet_AAA6AwAqGhAxMjUwND
c20TYw0DY5BbZxkkmfZUMmnKAsE_THFApESwu7
XSet_AAA6AwAq2AkxMjUwNDc20TYw0Dg4Bhk10Dii0U0QlaAbfG-dvMwBUxFk   XSet_AAA6AwAqGhAxMjUwND
c20TYw0DY5BbZxkkmfZUMmnKAsE_THFApESwu7.xml
XSet_AAA6AwAq2AkxMjUwNDc20TYw0Dg4Bhk10Dii0U0QlaAbfG-dvMwBUxFk.xml XSet_AAA6AwAq_jsxMjUwND
c20TYw0Tg0CuIRPc1YkUozqtuI-oCaeTkQx4n7
XSet_AAA6AwAq4JQxMjUwNDc20TYw0TI5CNTnRQ44okBPnnJFleb57UFNj5t1     XSet_AAA6AwAq_jsxMjUwND
c20TYw0Tg0CuIRPc1YkUozqtuI-oCaeTkQx4n7.xml
XSet_AAA6AwAq4JQxMjUwNDc20TYw0TI5CNTnRQ44okBPnnJFleb57UFNj5t1.xml XSet_AAA6AwAqJ0xMjUwND
c20TYw0DMxAxk6BG70QErMh3iqx3BMNwYGbeYJ
XSet_AAA6AwAq8QxMjUwNDc20TYw0TA3B3Mj-tqXN0YJjuftnzTjYW5uPlqR     XSet_AAA6AwAqJ0xMjUwND
c20TYw0DMxAxk6BG70QErMh3iqx3BMNwYGbeYJ.xml
XSet_AAA6AwAq8QxMjUwNDc20TYw0TA3B3Mj-tqXN0YJjuftnzTjYW5uPlqR.xml XSet_AAA6AwAqURQxMjUwND
c20TYw0TU1CTY-vTycKkMhgRj22XFJ-CpMdnKG
XSet_AAA6AwAqbH8xMjUwNDc20TYw0DExAjloAZmNdUUPto94Weh_11seuv3_     XSet_AAA6AwAqURQxMjUwND
c20TYw0TU1CTY-vTycKkMhgRj22XFJ-CpMdnKG.xml
XSet_AAA6AwAqbH8xMjUwNDc20TYw0DExAjloAZmNdUUPto94Weh_11seuv3_.xml XSet_AAA6AwAqUWoxMjUwND
c20TYw0DUxBM5rT72XVEEeuQvgMU1G47JugiVN
XSet_AAA6AwAqbu8xMjUwNDc20TYwNzcwAcSLfePaTULtNoaEkQ5kB0gDBBh2     XSet_AAA6AwAqUWoxMjUwND
c20TYw0DUxBM5rT72XVEEeuQvgMU1G47JugiVN.xml
XSet_AAA6AwAqbu8xMjUwNDc20TYwNzcwAcSLfePaTULtNoaEkQ5kB0gDBBh2.xml
$
```

Here is the contents of one of these XML files. It corresponds to one post in my blog. You will see the set of properties the application created for this particular post towards the end of the file.

```
1.0.0

org.snia.refvim.default.mgmt.policy

false

org.snia.refvim.default.mgmt.policy

true

2009-08-16T21:42:40.742-05:00

base
```

```
event

2009-08-16T21:42:40.742-05:00

2009-08-16T21:42:40.742-05:00

2009-08-16T21:42:40.724-05:00

2009-08-16T21:42:40.742-05:00

2009-08-16T21:42:40.742-05:00

AAA6AwAqbu8xMjUwNDc20TYwNzcwAcSLfePaTUlTnoaEkQ5kB0gDBBh2

fpmurphy

AWS, Java, XAM

http://blog.fpmurphy.com/2009/08/amazon-simpledb-typica-tutorial.html

2009-08-04T12:35:14.000-05:00

Amazon SimpleDB Typica Tutorial
```

Well, that is all for now. As usual, I hope this post was useful to you and that you have increased your knowledge of XAM Storage Systems and how to use the XAM Reference VIM.

As the SNIA XAM SDK matures, the Reference VIM is becoming more useful for prototyping a XAM Storage System. Maybe one day, road warriors can use something like the Reference VIM to do useful work while disconnected from their corporate XAM Storage System with the Reference VIM automatically syncing up with the corporate XAM Storage System when a secure network connection becomes available.