

JATS XML to CrossRef XML Conversion

Learning goal

- 1) Able to deposit DOI XML converted JATS 1.0 at <http://www.crossref.org/webDeposit/> with CrossRef ID and password.
- 2) Able to describe frame of NLM.JATS2CrossRef.v1.0.xsl XSL transformer

If JATS XML convert to Full text XML, It can create DOI XML easily.
It's easy to convert XML to PDF, Graph, VRML using trasnsformer.

NLM.JATS2CrossRef.v1.0.xsl

Source is as follows,

```
<?xml version="1.0"?>
<!-- Originally created by Aptara, Technology Group -->
<!-- Revised by CrossRef to accomodate NISO JATS 1.0 -->

<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.crossref.org/schema/4.3.1"
xmlns:xsldoc="http://www.bacman.net/XSLdoc"
xmlns:xlink="http://www.w3.org/1999/xlink"
exclude-result-prefixes="xsldoc">

<xsl:output method="xml"
indent="yes"
encoding="UTF-8"/>

<xsl:variable name="date" select="adjust-date-to-timezone(current-date(), ())"/>
<xsl:variable name="time" select="adjust-time-to-timezone(current-time(), ())"/>
<xsl:variable name="tempdatetime" select="concat($date,$time)"/>
<xsl:variable name="datetime" select="translate($tempdatetime,'-.','')"/>

<!-- ===== -->
<!-- Root Element -->
<!-- ===== -->
<xsl:template match="/">
<xsl:choose>
<xsl:when test="article">
<doi_batch version="4.3.1">
<xsl:attribute name="xsi:schemaLocation">http://www.crossref.org/schema/4.3.1
http://www.crossref.org/schema/deposit/crossref4.3.1.xsd</xsl:attribute>
<head>
<xsl:apply-templates select="//front"/>

</head>
```

```

<body>
<journal>
<xsl:apply-templates select="//journal-meta"/>
<xsl:if test="//pub-date||/article-meta/volume||/article-meta/issue">
<journal_issue>
<xsl:apply-templates select="//pub-date"/>
<xsl:apply-templates select="//article-meta/volume"/>
<xsl:apply-templates select="//article-meta/issue"/>
</journal_issue>
</xsl:if>
<xsl:apply-templates select="//article-meta/title-group"/>
</journal>
</body>
</doi_batch>
</xsl:when>
<xsl:otherwise>
<xsl:message terminate="yes"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- ===== -->
<!-- Front Matter Element -->
<!-- ===== -->
<xsl:template match="front">
<doi_batch_id>
<xsl:choose>
<xsl:when test="article-meta/article-id[@pub-id-type='art-access-id']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='art-access-id']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='publisher-id']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='publisher-id']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='doi']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='doi']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='medline']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='medline']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='pii']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='pii']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='sici']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='sici']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='pmid']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='pmid']"/>
</xsl:when>
<xsl:when test="article-meta/article-id[@pub-id-type='other']">
<xsl:apply-templates select="article-meta/article-id[@pub-id-type='other']"/>

```

```

</xsl:when>
<xsl:otherwise>
<xsl:comment>No article-id has been entered by user</xsl:comment>
</xsl:otherwise>
</xsl:choose>
</doi_batch_id>
<timestamp>
<xsl:value-of select="$datetime"/>
</timestamp>
<depositor>
<name>
<xsl:choose>
<xsl:when test="//journal-meta/publisher">
<xsl:apply-templates select="//journal-meta/publisher/publisher-name"/>
</xsl:when>
<xsl:otherwise>
<xsl:comment>Publisher's Name not found in the input file</xsl:comment>
</xsl:otherwise>
</xsl:choose>
</name>
<!--there is no appropriate place in NLM/JATS XML for a CrossRef deposit email, this will be added during processing (method TBD)-->
<email_address>TBD</email_address>
</depositor>
<registrant>
<xsl:choose>
<xsl:when test="//journal-meta/publisher">
<xsl:apply-templates select="//journal-meta/publisher/publisher-name"/>
</xsl:when>
<xsl:otherwise>
<xsl:comment>Publisher's name not found in the input file</xsl:comment>
</xsl:otherwise>
</xsl:choose>
</registrant>
</xsl:template>

<!-- ===== -->
<!-- Journal Metadata Element -->
<!-- ===== -->
<xsl:template match="journal-meta">
<journal_metadata language="en">
<xsl:choose>
<xsl:when test="journal-title-group/journal-title">
<full_title>
<xsl:value-of select="journal-title-group/journal-title"/>
</full_title>
</xsl:when>
<xsl:when test="journal-title">
<full_title>
<xsl:value-of select="journal-title"/>

```

```

</full_title>
</xsl:when>
<xsl:otherwise>
<full_title>
<xsl:message terminate="yes">Journal full title is not available in the Input file</xsl:message>
</full_title>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<xsl:when test="abbrev-journal-title">
<abbrev_title>
<xsl:value-of select="abbrev-journal-title"/>
</abbrev_title>
</xsl:when>
</xsl:choose>
<xsl:choose>
<xsl:when test="issn">
<xsl:apply-templates select="issn"/>
</xsl:when>
<xsl:otherwise>
<xsl:message terminate="yes">ISSN is not available in the Input file</xsl:message>
</xsl:otherwise>
</xsl:choose>
<xsl:if test="../article-meta/article-id[@pub-id-type='coden']">
<coden>
<xsl:value-of select="../article-meta/article-id[@pub-id-type='coden']"/>
</coden>
</xsl:if>
</journal_metadata>
</xsl:template>

<!-- ===== -->
<!-- ISSN -->
<!-- ===== -->
<xsl:template match="issn">
<xsl:if test="@pub-type='ppub'>
<issn media_type="print">
<xsl:apply-templates/>
</issn>
</xsl:if>
<xsl:if test="@pub-type='epub' or @pub-type='epub-ppub'">
<issn media_type="electronic">
<xsl:apply-templates/>
</issn>
</xsl:if>
<xsl:if test="not(@pub-type)">
<issn media_type="print">
<xsl:apply-templates/>
</issn>
</xsl:if>

```

```
</xsl:template>

<!-- ===== -->
<!-- Publication Date -->
<!-- ===== -->

<xsl:template match="pub-date">
<xsl:if test="@pub-type='ppub'>
<publication_date media_type="print">
<xsl:if test="month">
<month>
<xsl:apply-templates select="month"/>
</month>
</xsl:if>
<xsl:if test="day">
<day>
<xsl:apply-templates select="day"/>
</day>
</xsl:if>
<year>
<xsl:apply-templates select="year"/>
</year>
</publication_date>
</xsl:if>
<xsl:if test="@pub-type='epub' or @pub-type='epub-ppub'">
<publication_date media_type="online">
<xsl:if test="month">
<month>
<xsl:apply-templates select="month"/>
</month>
</xsl:if>
<xsl:if test="day">
<day>
<xsl:apply-templates select="day"/>
</day>
</xsl:if>
<year>
<xsl:apply-templates select="year"/>
</year>
</publication_date>
</xsl:if>
<xsl:if test="not(@pub-type)">
<publication_date media_type="print">
<xsl:if test="month">
<month>
<xsl:apply-templates select="month"/>
</month>
</xsl:if>
<xsl:if test="day">
<day>
<xsl:apply-templates select="day"/>
```

```

</day>
</xsl:if>
<year>
<xsl:apply-templates select="year"/>
</year>
</publication_date>
</xsl:if>
</xsl:template>

<!-- ===== -->
<!-- Volume/Issue -->
<!-- ===== -->
<xsl:template match="//article-meta/volume">
<journal_volume>
<volume>
<xsl:apply-templates/>
</volume>
</journal_volume>
</xsl:template>

<xsl:template match="//article-meta/issue">
<issue>
<xsl:apply-templates/>
</issue>
</xsl:template>

<!-- ===== -->
<!-- Title Group -->
<!-- ===== -->
<xsl:template match="//article-meta/title-group">
<journal_article publication_type="full_text">
<titles>
<title>
<xsl:apply-templates select="article-title"/>
</title>
</titles>
<xsl:if test="//article-meta/contrib-group">
<xsl:apply-templates select=".//contrib-group"/>
</xsl:if>
<xsl:apply-templates select="//pub-date"/>
<xsl:if test="//article-meta/fpage||//article-meta/lpage">
<xsl:apply-templates select="//article-meta/fpage||//article-meta/lpage"/>
</xsl:if>
<xsl:if test="//article-id[@pub-id-type='doi']||//article-id[@pub-id-type='pii']||//article-id[@pub-id-type='sici']">
<xsl:call-template name="publisher-item"/>
</xsl:if>
<doi_data>
<doi>
<xsl:choose>

```

```

<xsl:when test="//article-meta/article-id[@pub-id-type='doi']">
<xsl:apply-templates select="//article-meta/article-id[@pub-id-type='doi']"/>
</xsl:when>
<xsl:otherwise>
<xsl:message terminate="yes">DOI entry is not available in the Input/Meta file(s)</xsl:message>
</xsl:otherwise>
</xsl:choose>
</doi>
<timestamp>
<xsl:value-of select="$datetime"/>
</timestamp>
<resource>
<xsl:choose>
<xsl:when test="//article-meta/self-uri/@xlink:href">
<xsl:apply-templates select="//article-meta/self-uri/@xlink:href"/>
</xsl:when>
<xsl:otherwise>
<xsl:comment>No Resource entry has been entered by the user</xsl:comment>
</xsl:otherwise>
</xsl:choose>
</resource>
</doi_data>
<xsl:apply-templates select="//back/ref-list"/>
</journal_article>
</xsl:template>

<!-- ===== -->
<!-- Article Contributors -->
<!-- ===== -->
<xsl:template match="//article-meta/contrib-group">
<contributors>
<xsl:apply-templates select="contrib"/>
</contributors>
</xsl:template>

<xsl:template match="contrib">
<xsl:if test="position()=1">
<person_name sequence="first" contributor_role="author">
<xsl:apply-templates select="name"/>
<xsl:if test="xref[@ref-type='aff' and @rid]">
<xsl:call-template name="multi-ref">
<xsl:with-param name="tokens" select="xref[@ref-type='aff']/@rid"/>
</xsl:call-template>
</xsl:if>
</person_name>
</xsl:if>
<xsl:if test="position()>1">
<person_name sequence="additional" contributor_role="author">
<xsl:apply-templates select="name"/>
<xsl:if test="xref[@ref-type='aff' and @rid]">

```

```

<xsl:call-template name="multi-ref">
<xsl:with-param name="tokens" select="xref[@ref-type='aff']/@rid"/>
</xsl:call-template>
</xsl:if>
</person_name>
</xsl:if>
</xsl:template>

<xsl:template match="contrib-group/contrib/name">
<xsl:if test="given-names">
<given_name>
<xsl:apply-templates select="given-names"/>
</given_name>
</xsl:if>
<surname>
<xsl:apply-templates select="surname"/>
</surname>
<xsl:if test="suffix">
<suffix>
<xsl:apply-templates select="suffix"/>
</suffix>
</xsl:if>
</xsl:template>

<xsl:template name="multi-ref">
<xsl:param name="tokens"/>
<xsl:if test="$tokens">
<xsl:choose>
<xsl:when test="contains($tokens, ' ')>
<xsl:call-template name="one-ref">
<xsl:with-param name="token" select="substring-before($tokens, ' ')"/>
</xsl:call-template>
<xsl:call-template name="multi-ref">
<xsl:with-param name="tokens" select="substring-after($tokens, ' ')"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="one-ref">
<xsl:with-param name="token" select="$tokens"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:if>
</xsl:template>

<xsl:template name="one-ref">
<xsl:param name="token"/>
<affiliation>
<xsl:value-of select="//aff[@id=$token]"/>
</affiliation>

```

```

</xsl:template>

<xsl:template match="aff">
</xsl:template>

<xsl:template match="aff/label">
</xsl:template>

<!-- ===== -->
<!-- Article Page Information -->
<!-- ===== -->
<xsl:template match="article-meta/fpage">
<pages>
<first_page>
<xsl:apply-templates/>
</first_page>
<xsl:if test=".//lpage">
<last_page>
<xsl:value-of select=".//lpage"/>
</last_page>
</xsl:if>
</pages>
</xsl:template>

<xsl:template match="lpage">
</xsl:template>

<!-- ===== -->
<!-- Publication Identifier -->
<!-- ===== -->
<xsl:template name="publisher-item">
<publisher_item>
<xsl:if test="//article-id[@pub-id-type='doi']">
<identifier id_type="doi">
<xsl:value-of select="//article-id[@pub-id-type='doi']"/>
</identifier>
</xsl:if>
<xsl:if test="//article-id[@pub-id-type='pii']">
<identifier id_type="pii">
<xsl:value-of select="//article-id[@pub-id-type='pii']"/>
</identifier>
</xsl:if>
<xsl:if test="//article-id[@pub-id-type='sici']">
<identifier id_type="sici">
<xsl:value-of select="//article-id[@pub-id-type='sici']"/>
</identifier>
</xsl:if>
</publisher_item>
</xsl:template>

```

```
<!-- ===== -->
<!-- Citations -->
<!-- ===== -->
<xsl:template match="ref-list">
<citation_list>
<xsl:apply-templates select="ref"/>
</citation_list>
</xsl:template>

<xsl:template match="ref">
<xsl:variable name="key" select="concat($datetime,'_',@id)"/>
<citation>
<xsl:attribute name="key">key<xsl:value-of select="$key"/></xsl:attribute>
<xsl:apply-templates select="element-citation"/>
<xsl:apply-templates select="citation"/>
</citation>
</xsl:template>

<xsl:template match="element-citation | citation">
<xsl:choose>
<xsl:when test="@publication-type='journal' or @citation-type='journal'">
<xsl:if test="issn">
<issn>
<xsl:apply-templates select="issn"/>
</issn>
</xsl:if>
<xsl:if test="source">
<journal_title>
<xsl:apply-templates select="source"/>
</journal_title>
</xsl:if>
<xsl:if test="collab">
<xsl:apply-templates select="collab"/>
</xsl:if>
<xsl:if test="person-group">
<xsl:apply-templates select="person-group/name|person-group/collab"/>
</xsl:if>
<xsl:if test="volume">
<volume>
<xsl:apply-templates select="volume"/>
</volume>
</xsl:if>
<xsl:if test="issue">
<issue>
<xsl:apply-templates select="issue"/>
</issue>
</xsl:if>
<xsl:if test="fpage">
<first_page>
<xsl:apply-templates select="fpage"/>
```

```

</first_page>
</xsl:if>
<xsl:if test="year">
<cYear>
<xsl:value-of select="replace(year, '[a-zA-Z]', '')" />
</cYear>
</xsl:if>
<xsl:if test="article-title">
<article_title>
<xsl:apply-templates select="article-title"/>
</article_title>
</xsl:if>
</xsl:when>
<xsl:when test="@citation-type='book' or @citation-type='conf-proceedings' or @citation-
type='confproc' or @citation-type='other' or @publication-type='book' or @publication-type='conf-
proceedings' or @publication-type='confproc' or @publication-type='other'">
<xsl:if test="source">
<volume_title>
<xsl:apply-templates select="source"/>
</volume_title>
</xsl:if>
<xsl:if test="collab">
<xsl:apply-templates select="collab"/>
</xsl:if>
<xsl:if test="person-group">
<xsl:apply-templates select="person-group/name|person-group/collab"/>
</xsl:if>
<xsl:if test="edition">
<edition_number>
<xsl:apply-templates select="edition"/>
</edition_number>
</xsl:if>
<xsl:if test="fpage">
<first_page>
<xsl:apply-templates select="fpage"/>
</first_page>
</xsl:if>
<xsl:if test="year">
<cYear>
<xsl:value-of select="replace(year, '[a-zA-Z]', '')" />
</cYear>
</xsl:if>
<xsl:if test="article-title">
<article_title>
<xsl:apply-templates select="article-title"/>
</article_title>
</xsl:if>
</xsl:when>
<xsl:otherwise>
<unstructured_citation>

```

```

<xsl:value-of select=". "/>
</unstructured_citation>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<xsl:template match="back//name">
<xsl:if test="position()=1">
<author>
<xsl:apply-templates select="surname"/>
</author>
</xsl:if>
</xsl:template>

<xsl:template match="back//collab">
<xsl:if test="position()=1">
<author>
<xsl:apply-templates/>
</author>
</xsl:if>
</xsl:template>

</xsl:stylesheet>

```

when using this XSL, Should be recognized notice of JATS XML as blow.

1. Describe journal title to <journal-title element>.
2. Describe DOI to pub-id-type='doi', as attribute at <article-id>.
3. To input DOI URL to <self-uri>.
4. JATS DTD hasn't email of people who deposited CrossRef. So It should convert DOI XML in person, of course, It can be insert automatically after DOI XML, however; It should use XSL.

Create DOI XML in website.

1. Access <http://www.crossref.org/webDeposit/>.
2. Select Data Type
Select Data Type: oJournal oBook oConference Proceedings oReport ■NLM File BETA

The screenshot shows the 'Data Type Selection' step of the CrossRef webDeposit process. The 'NLM File BETA' option is selected. A note in the center of the form states: 'You need JATS XML, CrossRef ID and password'. The form includes fields for journal information like Title, Abbreviation, DOI, URL, ISBN, Volume, Issue, and publication dates. At the bottom are 'Submit Journal DOI' and 'Add Articles' buttons.

3. Send Email

From: CrossRef Query System [mailto:admin@crossref.org] Sent: Thursday, June 20, 2013 7:49 AM
<!-- Sender email --> To: shuh@hallym.ac.kr Subject: CrossRef submission ID: 1359730833<?xml version="1.0" encoding="UTF-8"?><doi_batch_diagnostic status="completed" sp="ds4.crossref.org"><submission_id>1359730833</submission_id><batch_id>jeehp-10-04</batch_id><record_diagnostic status="Success"><doi>10.3352/jeehp.2013.10.4</doi><msg>Successfully updated</msg><citations_diagnostic><citation key="key20130619184821825_b1-jeehp-10-04" status="resolved_reference">10.1016/j.jalz.2010.11.007</citation><citation key="key20130619184821825_b2-jeehp-10-04" status="resolved_reference">10.1007/s10072-003-0193-0</citation><citation key="key20130619184821825_b3-jeehp-10-04" status="resolved_reference">10.1111/j.1532-5415.2007.01249.x</citation><citation key="key20130619184821825_b4-jeehp-10-04" status="resolved_reference">10.7326/0003-4819-131-4-199908170-00002</citation><citation key="key20130619184821825_b5-jeehp-10-04" status="resolved_reference">10.1093/gerona/59.6.M621</citation><citation key="key20130619184821825_b6-jeehp-10-04" status="resolved_reference">10.1097/00002093-199706000-00006</citation><citation key="key20130619184821825_b7-jeehp-10-04" status="stored_query"></citation><citation key="key20130619184821825_b8-jeehp-10-04" status="stored_query"></citation><citation key="key20130619184821825_b9-jeehp-10-04" status="resolved_reference">10.1111/j.1532-5415.2005.00473.x</citation><citation key="key20130619184821825_b10-jeehp-10-04" status="resolved_reference">10.1177/147130120600500311</citation></citations_diagnostic></record_diagnostic><batch_data><record_count>1</record_count><success_count>1</success_count><warning_count>0</warning_count><failure_count>0</failure_count></batch_data></doi_batch_diagnostic>

So, It insert DOI to reference have JATS XML. and If it hasn't DOI at reference, Insert data which be extracted.

It send DOI file automatically as blow,

```
<?xml version="1.0" encoding="UTF-8"?>
<doi_batch xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.crossref.org/schema/4.3.1"
  version="4.3.1"
  xsi:schemaLocation="http://www.crossref.org/schema/4.3.1
  http://www.crossref.org/schema/deposit/crossref4.3.1.xsd">
  <head>
    <doi_batch_id>jeehp-10-04</doi_batch_id>
    <timestamp>20130619184821825</timestamp>
    <depositor>
      <name>National Health Personnel Licensing Examination Board of the Republic of Korea</name>
      <email_address>younsg@m2comm.co.kr</email_address>
    </depositor>
    <registrant>xmla</registrant>
  </head>
  <body>
    <journal>
      <journal_metadata language="en">
```

<full_title>Journal of Educational Evaluation for Health Professions</full_title>
<issn media_type="electronic">1975-5937</issn>
</journal_metadata>
<journal_issue>
<publication_date media_type="online">
<month>05</month>
<day>27</day>
<year>2013</year>
</publication_date>
<journal_volume>
<volume>10</volume>
</journal_volume>
</journal_issue>
<journal_article publication_type="full_text">
<titles>
<title>United States medical students' knowledge of Alzheimer disease</title>
</titles>
<contributors>
<person_name sequence="first" contributor_role="author">
<given_name>Brian J.</given_name>
<surname>Nagle</surname>
</person_name>
<person_name sequence="additional" contributor_role="author">
<given_name>Paula M.</given_name>
<surname>Usita</surname>
</person_name>
<person_name sequence="additional" contributor_role="author">
<given_name>Steven D.</given_name>
<surname>Edland</surname>
</person_name>
</contributors>
<publication_date media_type="online">
<month>05</month>
<day>27</day>
<year>2013</year>
</publication_date>
<publisher_item>
<identifier id_type="doi">10.3352/jeehp.2013.10.4</identifier>
</publisher_item>
<doi_data>
<doi>10.3352/jeehp.2013.10.4</doi>
<timestamp>20130619184821825</timestamp>
<resource><http://www.jeehp.org/DOIx.php?id=10.3352/jeehp.2013.10.4></resource>
</doi_data>
<citation_list>
<citation key="key20130619184821825_b1-jeehp-10-04">
<journal_title>Alzheimers Dement</journal_title>
<author>Brookmeyer</author>
<volume>7</volume>
<first_page>61</first_page>

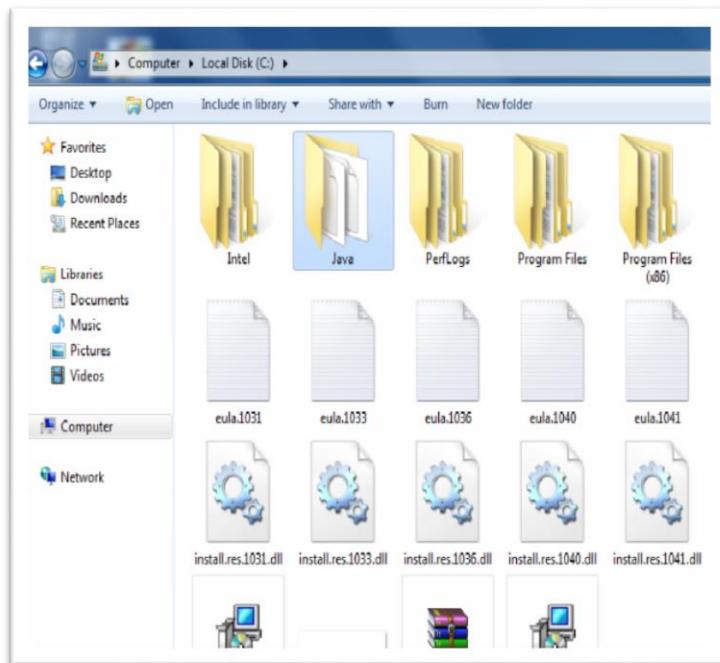
<cYear>2011</cYear>
<article_title>National estimates of the prevalence of Alzheimer's disease in the United States</article_title>
</citation>
<citation key="key20130619184821825_b2-jeehp-10-04">
<journal_title>Neurol Sci</journal_title>
<author>Pucci</author>
<volume>24</volume>
<first_page>384</first_page>
<cYear>2004</cYear>
<article_title>General practitioners facing dementia: are they fully prepared?</article_title>
</citation>
<citation key="key20130619184821825_b3-jeehp-10-04">
<journal_title>J Am Geriatr Soc</journal_title>
<author>Chodosh</author>
<volume>55</volume>
<first_page>1260</first_page>
<cYear>2007</cYear>
<article_title>Caring for patients with dementia: how good is the quality of care? Results from three health systems</article_title>
</citation>
<citation key="key20130619184821825_b4-jeehp-10-04">
<journal_title>Ann Intern Med</journal_title>
<author>Gifford</author>
<volume>131</volume>
<first_page>237</first_page>
<cYear>1999</cYear>
<article_title>Improving adherence to dementia guidelines through education and opinion leaders: a randomized, controlled trial</article_title>
</citation>
<citation key="key20130619184821825_b5-jeehp-10-04">
<journal_title>J Gerontol A Biol Sci Med Sci</journal_title>
<author>Boise</author>
<volume>59</volume>
<first_page>M621</first_page>
<cYear>2004</cYear>
<article_title>Dementia assessment in primary care: results from a study in three managed care systems</article_title>
</citation>
<citation key="key20130619184821825_b6-jeehp-10-04">
<journal_title>Alzheimer Dis Assoc Disord</journal_title>
<author>Barrett</author>
<volume>11</volume>
<first_page>99</first_page>
<cYear>1997</cYear>
<article_title>Knowledge about Alzheimer disease among primary care physicians, psychologists, nurses, and social workers</article_title>
</citation>
<citation key="key20130619184821825_b7-jeehp-10-04">
<volume_title>Dementias, including alzheimer's disease</volume_title>

```
<author>US Department of Health and Human Services</author>
<cYear>2013</cYear>
</citation>
<citation key="key20130619184821825_b8-jeehp-10-04">
<journal_title>Acad Psychiatry</journal_title>
<author>Goldstein</author>
<volume>23</volume>
<first_page>142</first_page>
<cYear>1999</cYear>
<article_title>A course in demetia for third-year medical students</article_title>
</citation>
<citation key="key20130619184821825_b9-jeehp-10-04">
<journal_title>J Am Geriatr Soc</journal_title>
<author>Struck</author>
<volume>53</volume>
<first_page>2007</first_page>
<cYear>2005</cYear>
<article_title>Effect of a mandatory geriatric medicine clerkship on third-year students</article_title>
</citation>
<citation key="key20130619184821825_b10-jeehp-10-04">
<journal_title>Dementia</journal_title>
<author>Morhardt</author>
<volume>5</volume>
<first_page>448</first_page>
<cYear>2006</cYear>
<article_title>Educating medical students on Alzheimer's disease and related disorders: an overview of the Northwestern University Buddy Program</article_title>
</citation>
</citation_list>
</journal_article>
</journal>
</body>
</doi_batch>
```

Convert JATS 1.0 XML to DOI XML in local.

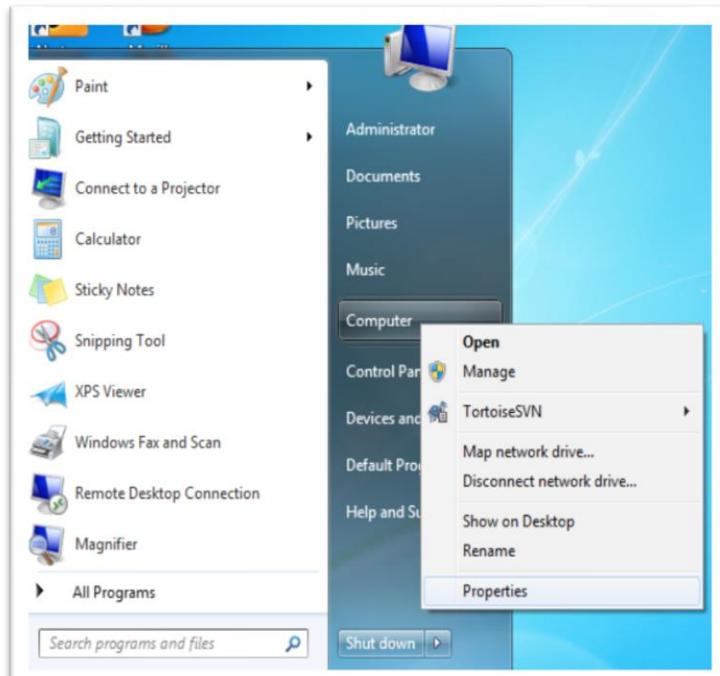
- 1) Able to install Java and saxon
- 2) Able to convert **Convert JATS 1.0 XML to DOI XML using NLM.JATS2CrossRef.v1.0.xsl**

Step 1: Where's my JDK (Java SE Development Kit) installation

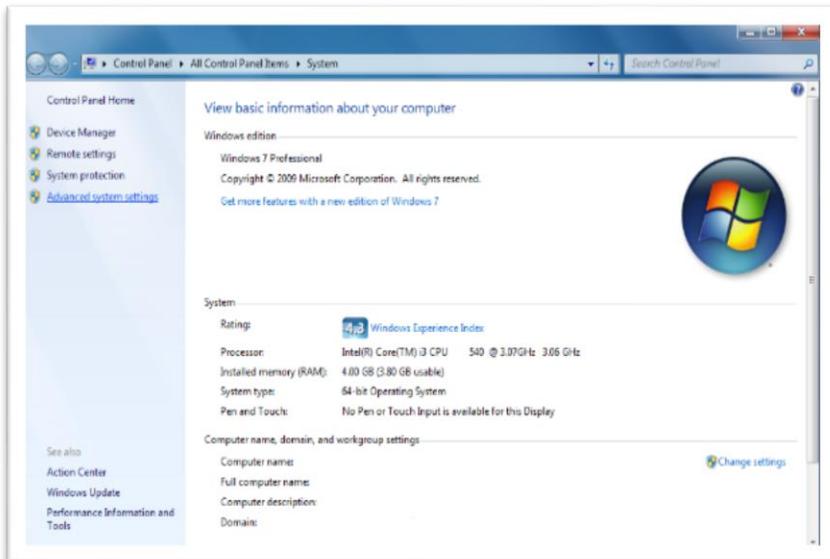


The first step would be to locate your jdk's installation folder on your hard drive. Mine was **C:\Java**. This is an important step. You need to know the jdk installation folder location. This will be required later.

Step 2: Setting up environment variable. Where do i begin?

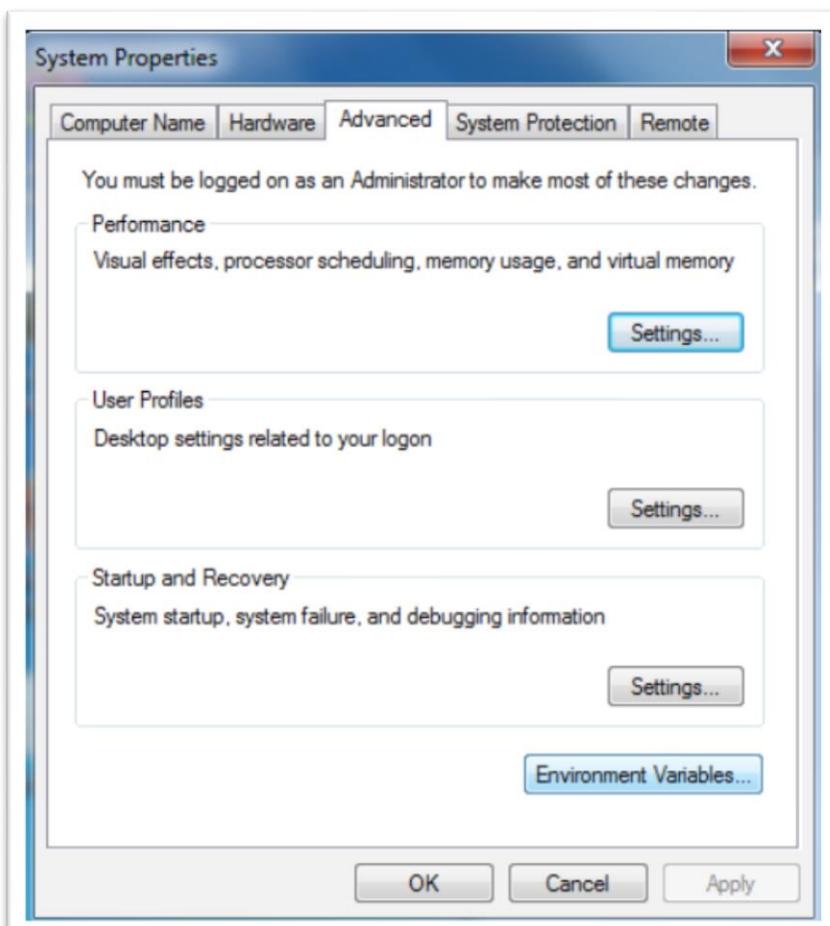


To set up the environment variable,right click on the 'My Computer'(or 'Computer' icon/menu item in Win 7)



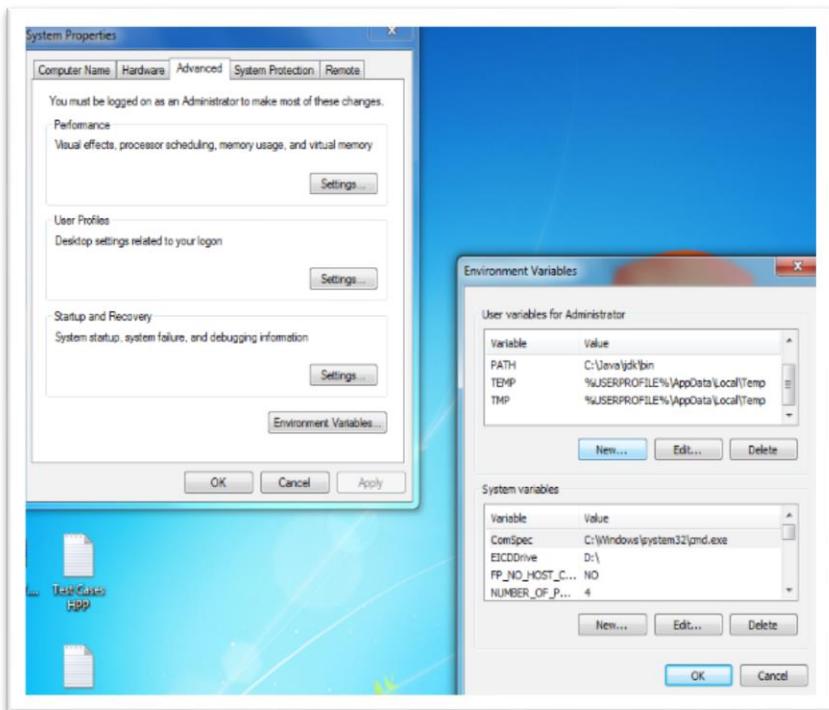
Notice the '**Advanced system settings**' on left!! Click on it.

This would throw up a new mini-window with the '**System properties**' options.



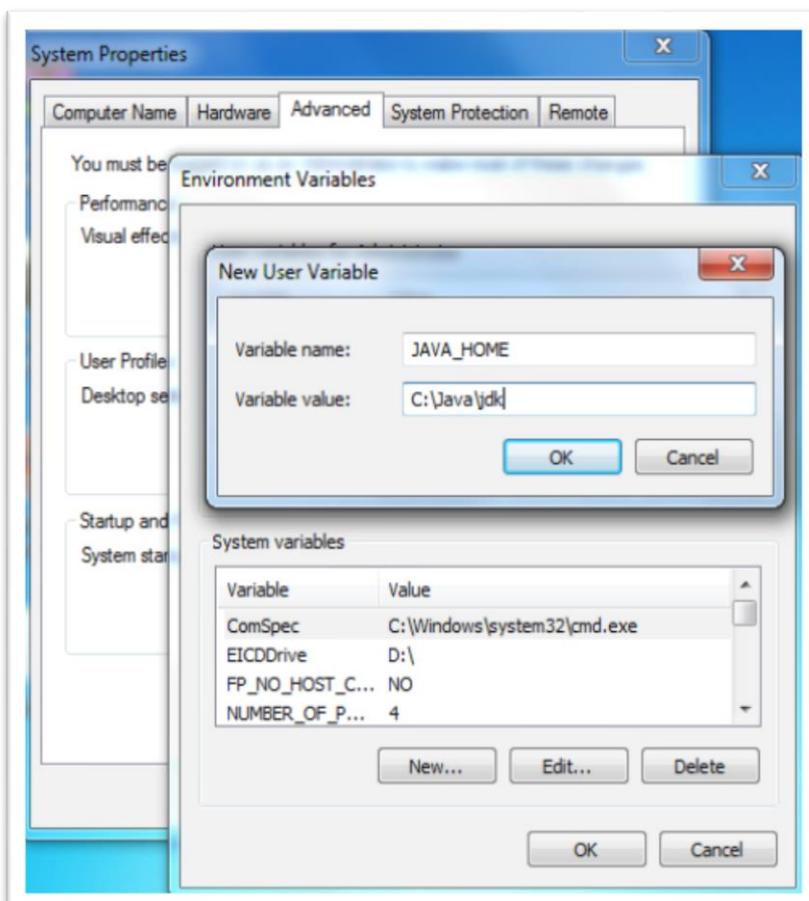
Now locate and click the button which says '**Environment Variables**'.

This shows up the EV window where you have a couple of system related variables pre-defined.

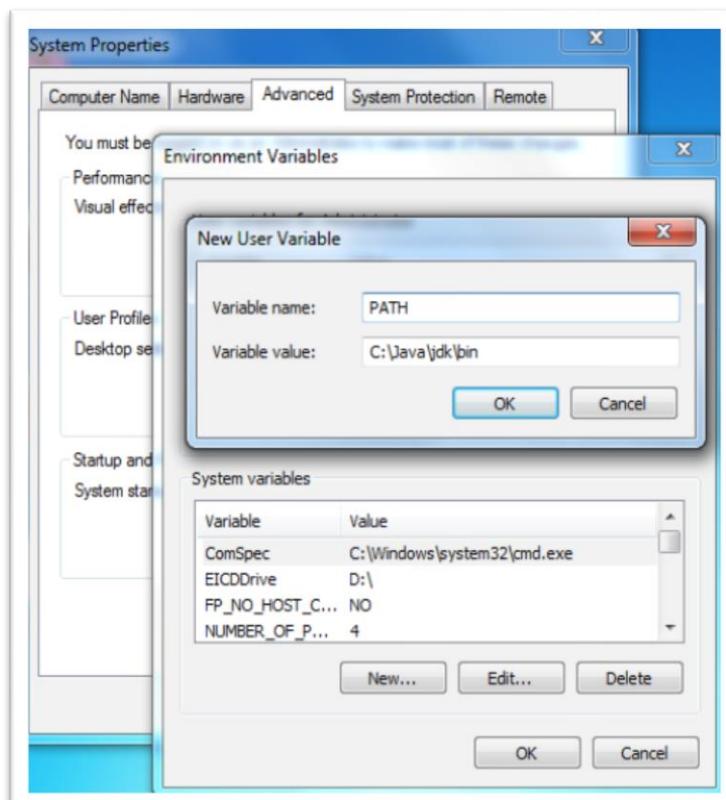


Step 3: Setting up things

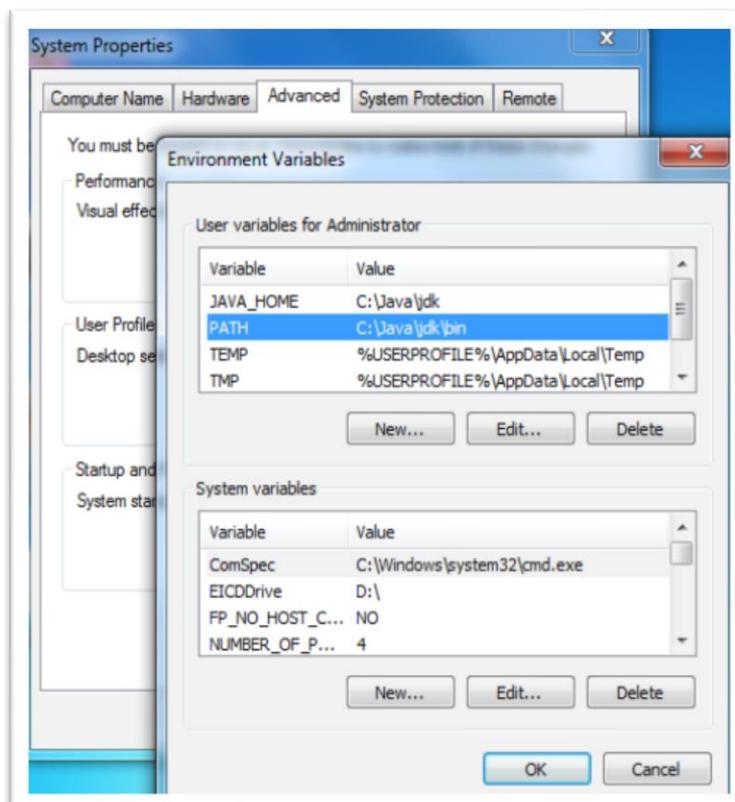
For **JAVA_HOME** put in these parameter values. Note: The **Variable value** should contain the location of the **JDK** folder on your system.



For PATH put in these values. Note: The **Variable value** should contain the location of the **bin** folder of the JDK installation on your system.

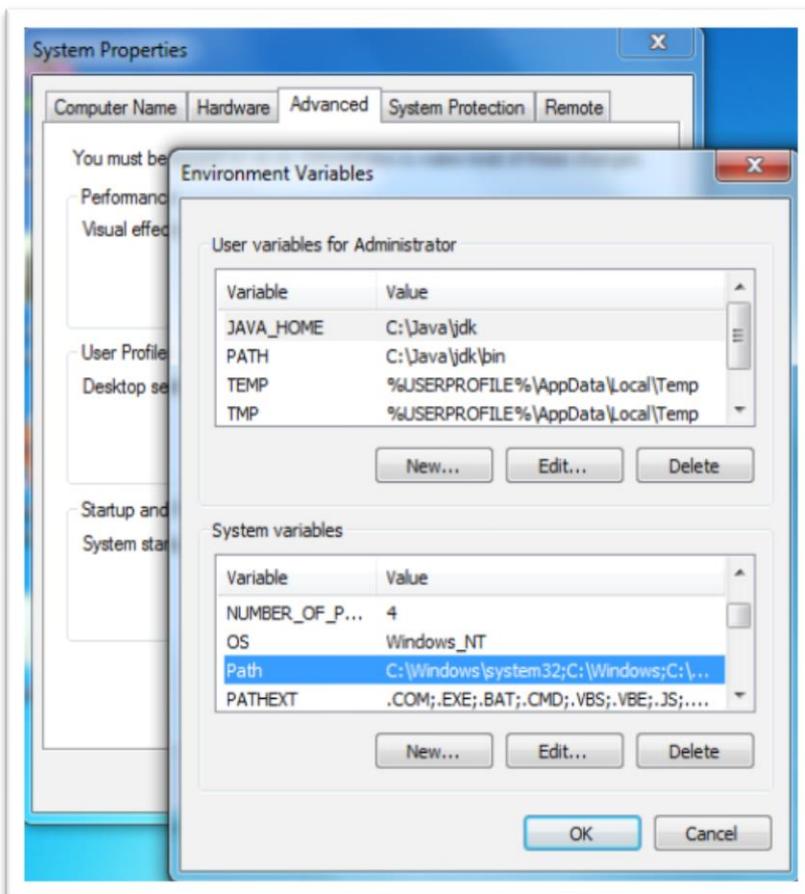


Once done, you can see these environment variables in the listing. But we are not done yet.



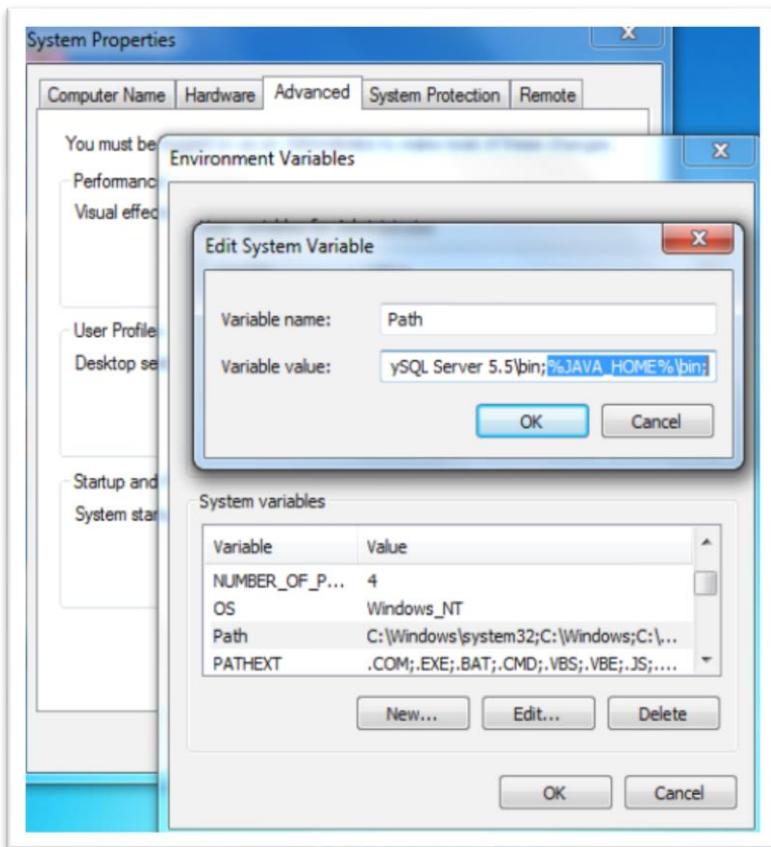
Step 4: Some more changes

We just specified the environment variables. But the job is not over yet. We need to edit some '**System Variables**' for the **JAVA_HOME** variable. To do so, highlight the **JAVA_HOME** variable by clicking on it. Once highlighted, you can see the various system variables related to our variable.



Now, we need to change the '**Path**' variable (highlighted in the image above). Click on it and then click on '**Edit**'.

This throws up the '**Edit System Variable**' mini-window. Now add the highlighted portion in the image shown below at the end. (Don't forget to put in the semi-colon).



Once done, close everything and restart your system for the changes to come into effect. Once restarted, go to the **command-line(cmd.exe)** and type-in the following commands: **java** and **javadoc**

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\>java -version
java version "1.6.0_18"
Java(TM) SE Runtime Environment (build 1.6.0_18-b07)
Java HotSpot(TM) Client VM (build 16.0-b13, mixed mode, sharing)

C:\>javac -version
javac 1.6.0_18

C:\>
```

You can see the version of Java installed on your system. This indicates that you have set the '**environment variables**' successfully on your system.

Conclusion

It is important JATS XML convert into DOI XML when it is inserted in CrossMark, ORCID and FundRef.

All editors should comprehend important of converted JATS XML and applied their works.

It is able to realize XML is not only in science field, but it is also realized in society of humanity field too.

It is not only to used in science academic journal area, but it is also possible to use in humanities and social sciences area too. It is able to realize through the book.

Therefore, it is also has to work right after choosing the professional XML complaint. Now, our country needs technique about the information processing published to change by different types of form such as XML to PDF type-setting, Graph, VRML and so on.

Our nation's academic journal area is not big enough to get investments.

The support of government develops a simple program to be used by academic journals and continuous get trains, and then the support of companies for the academic journal publics' standard will be grow in the future.

Reference

1. Huh S. Conversion of JATS 1.0 XML to DOI XML. In: Huh S editor. The 9th Editor's Workshop; 2013 Jul 4-5; Seoul: KCSE; 2013. p.85-98.
2. JAVA [Internet]. [cited by 2014 May 27]. Available from:
<http://www.oracle.com/us/technologies/java/overview/index.html>.
3. Tech talks [Internet]. [cited 2014.5.27]. Available from:
<http://bluesteel007india.blogspot.kr/2013/02/setting-up-javahome-and-path.html>