

# **LightLink XML and RSS Data Sources Installation Guide**



LightLink XML and RSS Data Sources  
Data Source Installation Guide

March 21, 2012

NOTICE OF TRADEMARKS:

Inova LightLink and its components are trademarks of Inova Solutions.

While reasonable efforts have been taken in the preparation of this document to ensure its accuracy, Inova Solutions, Inc. assumes no liability resulting from any errors or omissions in this manual, or from the use of the information contained herein.

© 2019 Inova Solutions, Inc.  
971 2nd ST S.E.  
Charlottesville, VA 22902  
434.817.8000  
[www.inovasolutions.com](http://www.inovasolutions.com)

## Table of Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Data Capabilities</b>	<b>1</b>
2.1. Data Fields	1
2.2. Limitations	2
2.3. What the Customer Provides	2
2.4. Hardware / Software Requirements	2
2.5. Compatibility	2
2.6. Licensing	2
<b>3. Installation</b>	<b>3</b>
3.1. Interfacing XML with LightLink Input Manager	3
3.2. Configuring the XML Data Source Properties	6
3.3. XML Transformation Test Results	7
3.4. Creating a Custom Style Sheet	8
<b>4. Best Practices</b>	<b>9</b>
<b>5. Troubleshooting</b>	<b>9</b>

## 1. Introduction

The LightLink XML Data Source Manager acquires data from an XML (Extensible Markup Language) data source.

**XML** (*eXtensible Markup Language*) is a simple but widely used database format most often generated from Web-based applications. XML is an outgrowth of **HTML** (*HyperText Markup Language*) which was designed primarily to *display* information rather than *store* it. Both are plain-text formats so you can view and edit them without special software, although using an editor that recognizes their key words, structures and extensions makes for much easier development, especially when confronted with large files. XML provides you with flexible and efficient information presentation and transfer features using standard web servers and browsers.

**XSL** (*eXtensible Style Language*) is a set of instructions, also a plain-text format and similar in syntax and structure to XML, that *transforms* the data contained in an XML file into other data formats.

There are 3 different methods to inputting XML data:

- The input to the DSMXML can be the standard Inova XML format that is generated by the Inova Solutions LightLink Data Publisher.
- The DSMXML also supports the ability to read any conforming XML document if you supply an appropriate style sheet (XSL file). Creating style sheets is beyond the scope of this manual, but there are many books and resources on the web that will help. You can also use the supplied XSL files as examples.
- Rich Site Summary or Really Simple Syndication (RSS) is a type of XML format that is used for syndicating Web data. ATOM is another type of XML file that is popular for syndicating Web data. Inova supplies style sheets for both of these formats. Therefore, Inova Solutions' LightLink XML component allows anyone to publish any content by creating an RSS or ATOM feed that the data source can locate. These feeds can be internet sites, network files or local files.

## 2. Data Capabilities

### 2.1. Data Fields

There are no standard data fields but always database specific ones. That said, an RSS Data Source Manager (DSM) will typically return 'RSS\_All' which will

combine all individual items in the feeds into a single data field for ease of message creation.

Data field typecasting: strings, integers, doubles. There are no date, time, realtime or currency formats.

## 2.2. Limitations

You will need to balance the load between data source / output channels and data update frequency, just as you would for any other LightLink DSM.

Note that a typical weather RSS will only need updates every five minutes as opposed to real time ACD monitoring, which requires much more frequent updates. For real time monitoring, a good rule of thumb is to at least double the 'Test DSM' time (explained later in this guide).

## 2.3. What the Customer Provides

The customer will provide either an XML file and location or a URL for the XML file.

## 2.4. Hardware / Software Requirements

The hardware and software requirements are the same as general LightLink server requirements. MSXML.dll is the core library LightLink will employ for its parsing operations; it is typically already present on a Windows machine. For further information on this library, visit the [List of Microsoft XML Parser \(MSXML\) versions](#).

## 2.5. Compatibility

XML data versioned "1.0" or higher.

## 2.6. Licensing

The LightLink License Key will require XML as a data source.

### 3. Installation

#### 3.1. Interfacing XML with LightLink Input Manager

The DSM for LightLink **\*requires\*** XSL instructions to transform XML data using the following two keywords: **Name** and **Value**. These give instructions for LightLink DSM to create data fields and maintain their contents.

*In the current incarnation of LightLink, the XML interface can create only a flat table structure; it does not support Data Grouping.* For instance, a Call Center queue cannot have attendant data *under* the queue, but rather must live within all data for all queues and depend upon the data field name for distinction. For instance, Figure 1 is an example of a typical LightLink ODBC DSM structure:

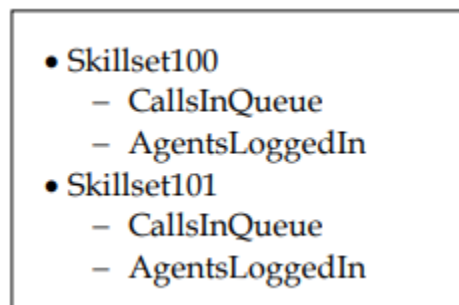


Figure 1

Figure 2 is the structure resulting from a LightLink XML DSM:

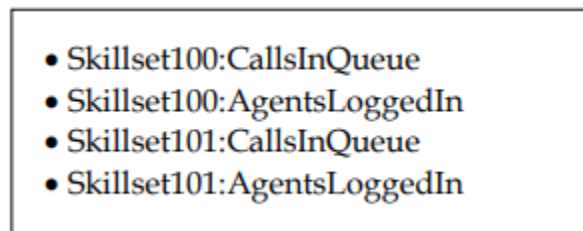


Figure 2

Inova's Product Management has received requests to enhance the XML DSM capabilities to include a new keyword **Group** to emulate other LightLink data source interfaces. A grouping allows for more user-friendly and logical access to groups of data and diminishes the issue of LightLink data field length limitation of 30 characters. (The section after the last ':' is the data field length.)

Consider the data fields in Figure 2 due to having to carry the group name in the data field name:

- ODBC:Skillset100:CallsInQueue (length=12)
- XML:Analyzer28:Skillset100-CallsInQueue (length=24)

XML interfacing with LightLink is typically a custom programming job as customers' XML database structures will be purpose-built for their individual requirements. Inova provides a unique value proposition to our customers' custom data; however, this presents the need for qualified XML/XSL programmers supporting this business model.

Figure 3 is an example of a simple XML file:

```

<information>
  <date>Tue May 10 13:16:42 PDT 2011</date>
  <channel>chat</channel>
  <header>
    <statheader id="0">Workgroup Name</statheader>
    <statheader id="2">Logged In</statheader>
    <statheader id="4">Available Agents</statheader>
    <statheader id="11">Interaction Offered</statheader>
    <statheader id="12">Queued Interactions</statheader>
    <statheader id="13">Answered Current</statheader>
    <statheader id="16">Interactions Abandoned</statheader>
    <statheader id="22">Current SL</statheader>
    <statheader id="30">Chat ASA</statheader>
    <statheader id="31">Chat Max Wait Time</statheader>
    <statheader id="35">Chat AHT</statheader>
    <statheader id="24">Phone ASA</statheader>
    <statheader id="25">Phone Max Wait Time</statheader>
    <statheader id="29">Phone AHT</statheader>
  </header>
  <statistics>
    <item id="Skill4">
      <statistic id="0">125</statistic>
      <statistic id="2">0</statistic>
      <statistic id="4">0</statistic>
      <statistic id="11">99</statistic>
      <statistic id="12">784</statistic>
      <statistic id="13">0</statistic>
      <statistic id="16">-45.5</statistic>
      <statistic id="22">100</statistic>
      <statistic id="30">01:02:45</statistic>
      <statistic id="31">0</statistic>
      <statistic id="35">0</statistic>
    </item>
    <item id="Skill191">
      <statistic id="0">1</statistic>
      <statistic id="2">2</statistic>
      <statistic id="4">3</statistic>
      <statistic id="11">4</statistic>
      <statistic id="12">5</statistic>
      <statistic id="13">6</statistic>
      <statistic id="16">7</statistic>
      <statistic id="22">8</statistic>
      <statistic id="30">00:05:56</statistic>
      <statistic id="31">10</statistic>
      <statistic id="29">11</statistic>
    </item>
  </statistics>
</information>
    
```

**Figure 3**

Figure 4 is a simple XSL on the previous XML data:

```

<!-- Inova Corp - Christian de la Pointe - May 25, 2011 -->
▼<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="text"/>
  <xsl:strip-space elements="*/>
  ▼<xsl:template match="information">
    Name: Date Value:
    <xsl:value-of select="date"/>
    Name: Channel Value:
    <xsl:value-of select="channel"/>
    <xsl:apply-templates select="statistics"/>
  </xsl:template>
  ▼<xsl:template match="statistics">
    ▼<xsl:for-each select="item">
      <xsl:variable name="WG_name" select="statistic[@id='0']"/>
      ▼<xsl:for-each select="statistic">
        Name:
        <xsl:value-of select="$WG_name"/>
        -
        ▼<xsl:call-template name="df_name">
          <xsl:with-param name="x" select="@id"/>
        </xsl:call-template>
        Value:
        <xsl:value-of select="."/>
      </xsl:for-each>
    </xsl:for-each>
  </xsl:template>
  ▼<xsl:template name="df_name">
    <xsl:param name="x"/>
    ▼<xsl:for-each select="/information/header/statheader">
      ▼<xsl:variable name="statid">
        <xsl:value-of select="@id"/>
      </xsl:variable>
      ▼<xsl:if test="$x=$statid">
        <xsl:value-of select="."/>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

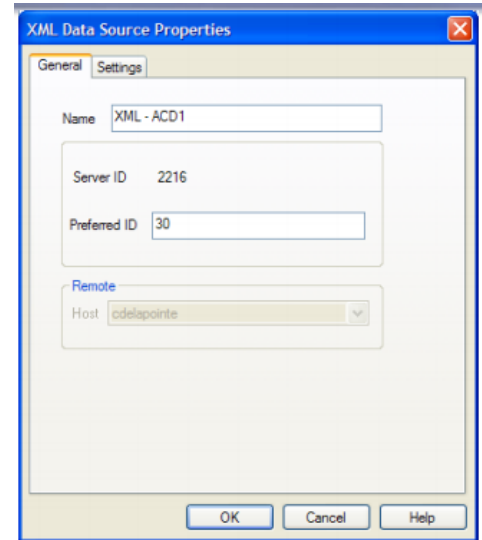
Figure 4



### 3.2. Configuring the XML Data Source Properties

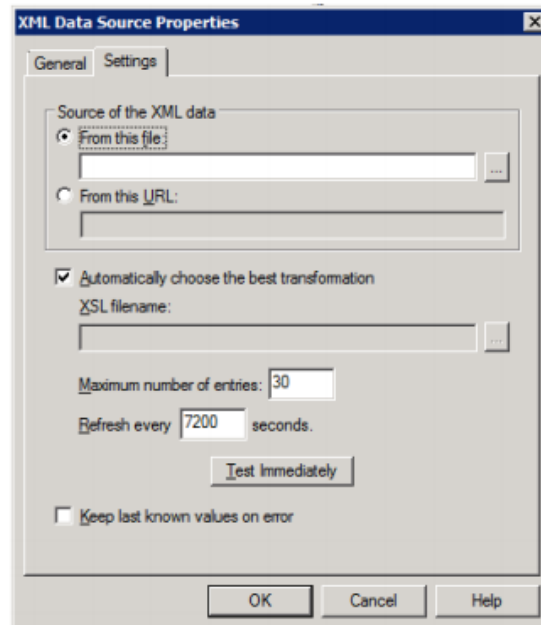
Once you have obtained the appropriate XSL for any given XML data source, the next step is to configure the XML Data Source Properties:

1. In LightLink Administrator, establish [config permissions]
2. Right-click Input Manager and select *New Data Source Connection...*
3. In the New Data Source dialog, select *XML Data Source* and click *OK*.
4. In the Properties dialog, enter the name of your new XML DSM (Figure 5). Notice that the *Name*, *Server ID*, *Preferred ID*, and *Host* are pre-populated. You can change the *Name*, *Preferred ID*, and *Host*.
5. Complete the following fields in the Settings tab of the Properties dialog (Figure 6):



**Figure 5**

- Select the *source of the XML data*, which will be either a file XML or a URL XML. *RSS will typically be a URL*. For this data source to find your data, you need to enter the file name or URL where your application is storing it.
- Select *Automatically choose the best transformation* to allow LightLink to determine the best match for the XML input file. Inova Solutions provides several XSL files that serve general purposes (e.g., an XSL file that understands Inova Solutions' internal XML Data Table and others that understand various RSS and ATOM formats).
- If desired, deselect the check box and enter the path of the custom XSL to use. The subfolder `server\srcvfg\dsm`s exists under the folder



**Figure 6**

where you installed your LightLink system. This subfolder holds the XSL files that are provided by Inova Solutions. You can create XSL files and add them to this folder by using the `.xsl` extension and naming them with the root node of the XML document.

- An example for the RSS feed (style sheet) is: `RSS.xsl` The beginning of every RSS feed has lines similar to:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
```

The first tag is `rss` so the style sheet is named `rss.xsl`. If you installed Inova Solutions LightLink to the default folder

```
C:\Program Files\Inova Solutions
```

then you will find the following file used as the default style sheet when the XML document is in RSS format:

```
C:\Program Files\Inova
Solutions\server\srcvfg\dsms\rss.xsl
```

- Enter a *maximum number of entries* to limit the number of data fields that will be reported. When a field is discovered and published it is added to the list of fields in LightLink Administrator and is available to the rest of the system. Fields are never removed so if you do not limit the number of entries published, the Inova Solutions LightLink system will have those entries until you manually delete them.
- Enter a number to specify to *refresh every \_\_ seconds*; this indicates how often LightLink will check for changes to your data.
- Click the *Test Immediately* button to ensure that the settings you have chosen are legal and that the data that will be published is what you want. This assumes that the XML document you have specified is currently available. (Refer to Section 3.3 for more information about test results.)

6. Click *OK* to save the changes.

### 3.3. XML Transformation Test Results

When you select the *Test Immediately* button the XML Transformation Test Results screen appears. The test attempts to read the XML document just as the running system will. A screen will appear with the results of this test; use this screen to determine if you typed the file and URL correctly, if this is a legal type of file, and if the XSL (style sheet) is accessible and correct.

If the test is successful, the data that will be published is displayed on this dialog. If the test is not successful, then as much error information as possible is displayed. Figure 7 shows an unsuccessful test.

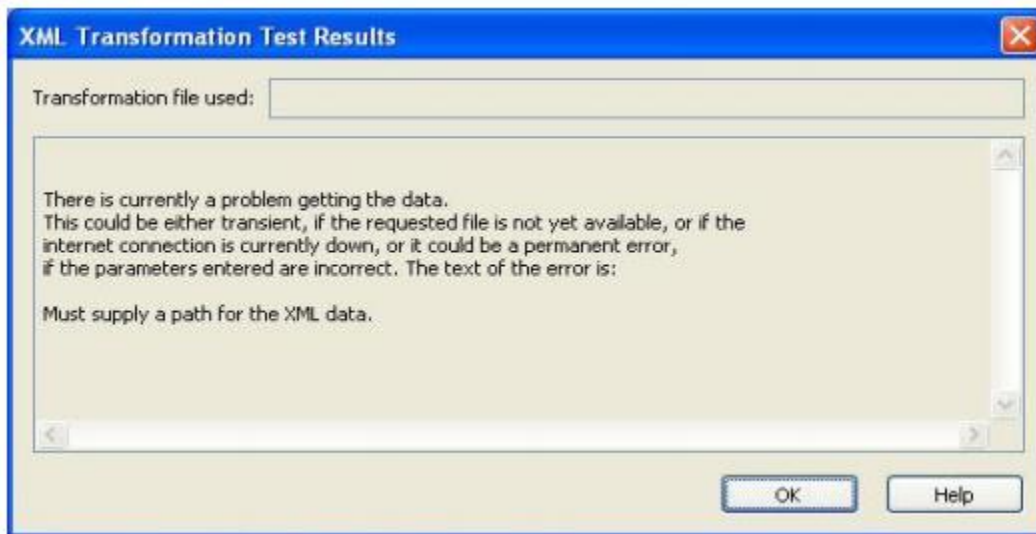


Figure 7

### 3.4. Creating a Custom Style Sheet

There are some situations where you will need to supply your own XSL file:

- You have a standard feed but you want some behavior other than the default behavior. (For instance, you want to skip an item or you want to control what the name of the field is.)
- You want to read an XML document for which Inova Solutions has not provided a style sheet.

To write a style sheet you will have to be familiar with XSLT; we suggest you look at the XSL files that are supplied by Inova Solutions for examples. The output of the transformation needs to fulfill certain requirements to allow LightLink to parse the feed data:

First line can be anything as long as it is not empty/blank; it is ignored.

The rest of the output needs to be pairs of lines in the following format:

```
Name: FIELD_NAME01
Value: item
```

*There is a mandatory space after the colons.*

FIELD\_NAME01 is the name of the field that gets published and item is the value of that field. LightLink will interpret the data in the most restrictive format possible. For instance, if you output:

```
Transformation arbitrary needed text
Name: FIELD_NAME01
Value: 50
Name: FIELD_NAME02
Value: Now is the time
```

The first field is called `FIELD_NAME01` and is considered a 16 bit integer. The second field is `FIELD_NAME02` and is considered a string.

## 4. Best Practices

See Section 2 of this guide for important data considerations. All other requirements to be evaluated by the Inova XML/XSL programmer.

## 5. Troubleshooting

Check the results of the **Test Immediately** operation. If it appears the XML data is not being even evaluated for the XSL, it may be either an Input Manager failure or a damaged MSMXL environment: restart the LightLink server and repair the MSXML (Windows Update, direct library replacement, see [MSDN MSXML](#)).

For further XSL troubleshooting, contact your Inova XML/XSL programmer.