

INDS EXECUTION MANAGER USER'S MANUAL (V2.0)

Prepared by:
Matt Miller and John Wilson
Erigo Technologies LLC
64 Main Street, P.O. Box 899
Enfield, NH 03748-0899
matt.miller@erigo.com

Submitted: October 21, 2010

This effort is sponsored by NASA Dryden Flight Research Center under
Contract NAS4-00047/NNDO7AB92C and
Arcata Associates Purchase Order Nbr 08DFR-0160

Document Change History

Version Number	Description of Change	Date Published	Author
V1.0	Original document	May 6, 2009	MJM/JPW
V2.0	Updates to Appendix A (installation instructions), mostly to reflect the use of Tomcat 6.	October 21, 2010	JPW

Introduction

The INDS Execution Manager (IEM) supports a platform-independent method of launching, monitoring, and terminating INDS applications such as the DataTurbine server, HttpMonitor, UDPCapture, and XMLDemux. IEM supports the basic INDS folder hierarchy (with Capture, Jobs, Projects, Server, etc. folders) or any other desired INDS folder system. Distinct from the platform-specific script-based INDS launch system (which uses a series of task-specific script files) the IEM maintains all startup command information in a single, XML file which must comply with a supplied IEM schema. This XML file is read and validated by the IEM at launch. IEM goes beyond the script-based INDS launch system by providing a means to conveniently access system health data and view system data flow, as shown in the “indsViewer” interface, accessible from a web browser (Figure 1). As such, IEM allows easy system debugging by non-INDS experts.

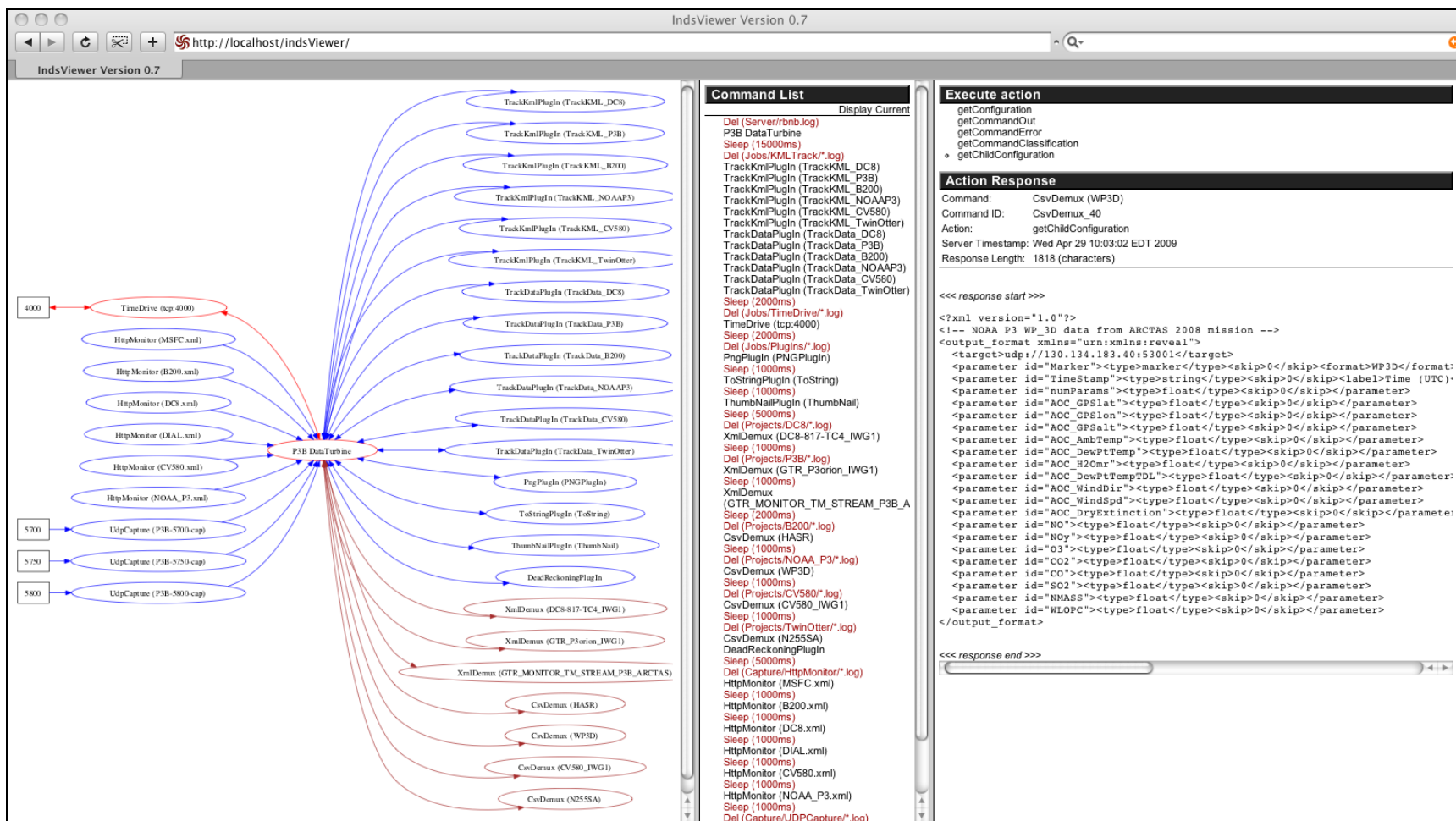


Figure 1. INDS Execution Manager “indsViewer” Interface.

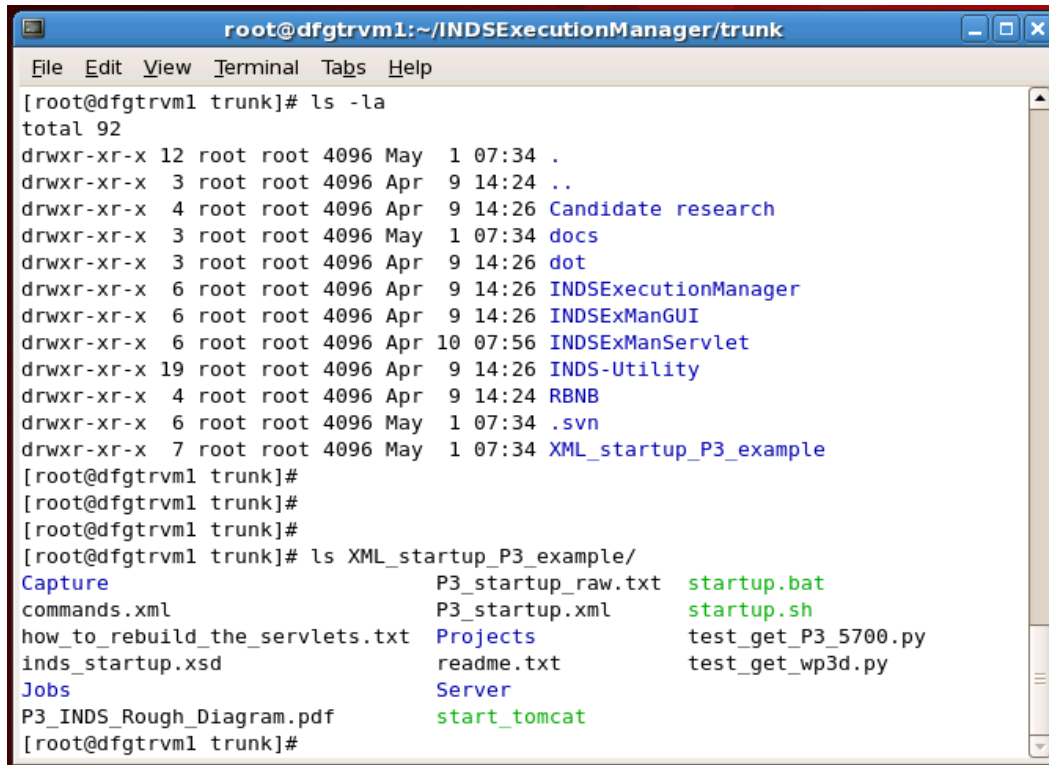
Installation and Setup

Detailed instructions for installation and setup of the INDS Execution Manager are included in Appendix A. The core IEM components are stored in a Subversion repository hosted by Google Code (<http://nasa-dryden-dt.googlecode.com/svn/trunk>). Briefly, installation requires the following steps: install prerequisite software (Java, Python, GraphViz, and Subversion), checkout a copy of the IEM files from the Google Code repository, and make minor changes to a few files to setup appropriate paths and environment variables.

Checking out a copy of the IEM files from Google Code will create the file hierarchy shown in Figure 2. This figure shows the content of *trunk* (the top folder checked out of the repository) as well as *trunk/XML_startup_P3_example*. The main folders of interest within trunk are:

- docs: contains IEM reference documentation – the IEM User’s Manual (this document) and the IEM Reference Manual.
- INDSExecutionManager: source code and binaries to run the core IEM functions.
- INDSExManGUI: source code and binaries for a client which connects to the IEM server and displays status information.
- INDSExManServlet: source code and binaries for a web client which connects to the IEM server and displays status information (this is the *indsViewer* interface).
- INDS-Utility: Repository of INDS utility applications such as UDPCapture and XMLDemux.
- RBNB: Directory for storing the RBNB distribution (currently contains V3.2B1). The Tomcat web/servlet server is included in this distribution.
- XML_startup_P3_example: an example INDS system (the NASA P3 system used during ARCTAS 2008). The content of this directory is also displayed in Figure 2. It contains the following:
 - The standard INDS folders: Capture, Jobs, Projects, Server
 - Files which describe the INDS system to be launched; these are read by IEM at launch: P3_startup.xml (a description of the P3 INDS system), commands.xml (details about the commands that can be executed), inds_startup.xsd (the schema to which P3_startup.xml must comply).
 - Windows and Linux script files for launching the IEM: start_tomcat, startup.sh, startup.bat.

- Test scripts to send raw data to the running P3 INDS system:
test_get_P3_5700.py and test_get_wp3d.py.



```

root@dfgtrvm1:~/INDSExecutionManager/trunk
File Edit View Terminal Tabs Help
[root@dfgtrvm1 trunk]# ls -la
total 92
drwxr-xr-x 12 root root 4096 May  1 07:34 .
drwxr-xr-x  3 root root 4096 Apr  9 14:24 ..
drwxr-xr-x  4 root root 4096 Apr  9 14:26 Candidate research
drwxr-xr-x  3 root root 4096 May  1 07:34 docs
drwxr-xr-x  3 root root 4096 Apr  9 14:26 dot
drwxr-xr-x  6 root root 4096 Apr  9 14:26 INDSExecutionManager
drwxr-xr-x  6 root root 4096 Apr  9 14:26 INDSExManGUI
drwxr-xr-x  6 root root 4096 Apr 10 07:56 INDSExManServlet
drwxr-xr-x 19 root root 4096 Apr  9 14:26 INDS-Utility
drwxr-xr-x  4 root root 4096 Apr  9 14:24 RBNB
drwxr-xr-x  6 root root 4096 May  1 07:34 .svn
drwxr-xr-x  7 root root 4096 May  1 07:34 XML_startup_P3_example
[root@dfgtrvm1 trunk]#
[root@dfgtrvm1 trunk]#
[root@dfgtrvm1 trunk]#
[root@dfgtrvm1 trunk]# ls XML_startup_P3_example/
Capture                               P3_startup_raw.txt      startup.bat
commands.xml                         P3_startup.xml          startup.sh
how_to_rebuild_the_servlets.txt      Projects                 test_get_P3_5700.py
inds_startup.xsd                     readme.txt               test_get_wp3d.py
Jobs                                 Server
P3_INDS_Rough_Diagram.pdf            start_tomcat
[root@dfgtrvm1 trunk]#

```

Figure 2. INDS Execution Manager file hierarchy and file list from the XML_startup_P3_example subdirectory.

Startup and Shutdown

Startup

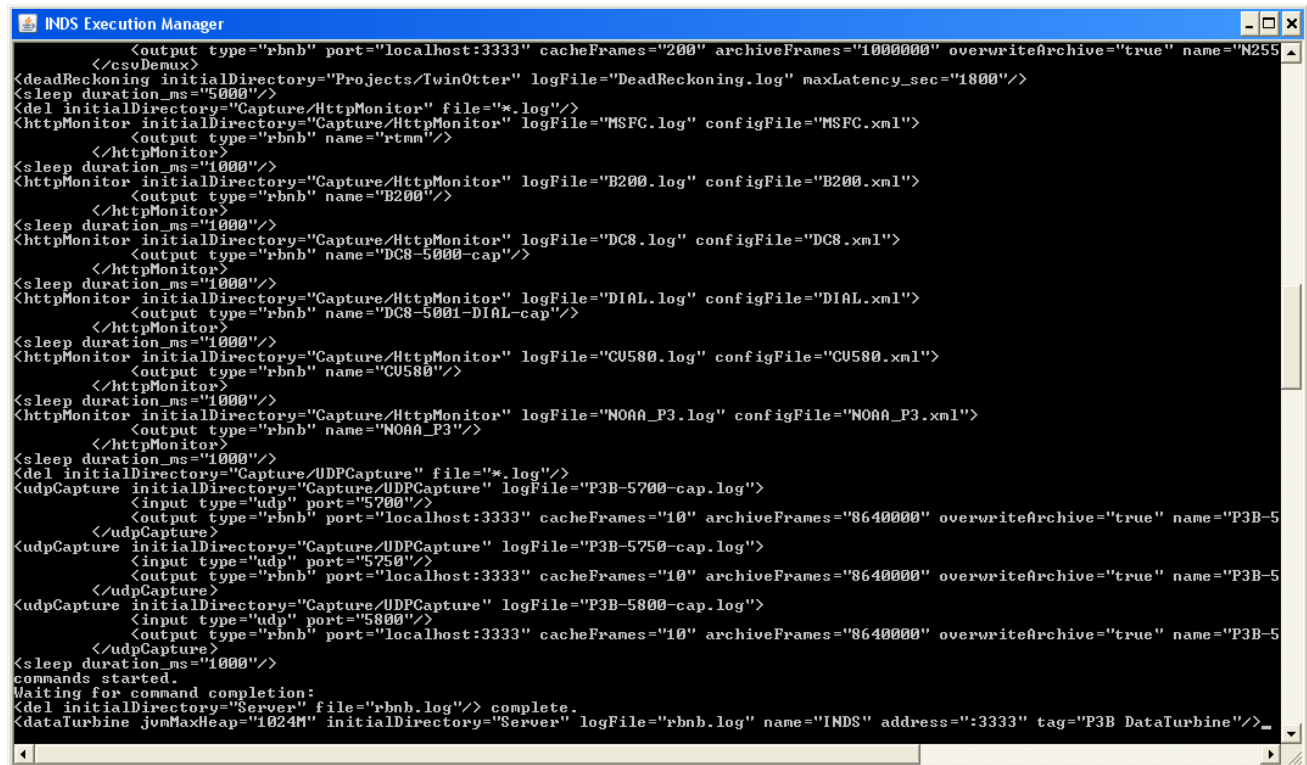
Once installation is complete, the included P3 INDS scenario (located in XML_startup_P3_example) can be executed. This sample scenario can be used to verify that your setup is working correctly. It can also be used as a template for creating your own custom INDS scenario.

To startup the P3 scenario:

1. Bring up a terminal/command window and go to the XML_startup_P3_example directory.
2. Depending on your platform, startup is performed in the following manner:
 - a. On Windows: execute *startup.bat*; this will start both the Tomcat server as well as IEM.
 - b. On Linux/Mac OS X:
 - i. Start the Tomcat server by executing *start_tomcat* as root. Root permission is required to launch Tomcat since the default HTTP port, 80, is used.
 - ii. Launch IEM by executing *startup.sh*.

Although it is possible to launch Tomcat from within IEM, we have chosen to start this server as a separate process. (Given the instructions above, this may not be apparent under Windows because only one script is executed. Within this script, however, Tomcat and IEM are separately executed.) The main reason for this is to appropriately support Linux/Mac OS X installations, where launching Tomcat to use port 80 requires root access. Thus, Tomcat is launched as root, and IEM is separately launched by a user with non-root permission.

As IEM is executing, debug messages are displayed in the terminal window (see Figure 3 for an example). This provides some feedback that the system is operating correctly. Several other steps can be taken to verify the system operation, as detailed in Appendix B. If everything is executing properly, you should be able to bring up the “indsViewer” interface in a web browser by going to <http://localhost/indsViewer> (for information about using this interface, see the next section, *Operation*). Using the “indsViewer” interface, it is a good idea to look through the “CommandOut” and “CommandError” content for all processes to make sure there are no obvious errors.



```

<output type="rbnb" port="localhost:3333" cacheFrames="200" archiveFrames="1000000" overwriteArchive="true" name="N255
</ssvDemux>
<deadReckoning initialDirectory="Projects/TwinOtter" logFile="DeadReckoning.log" maxLatency_sec="1800"/>
<sleep duration_ms="5000"/>
<del initialDirectory="Capture/HttpMonitor" file="*.log"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="MSFC.log" configFile="MSFC.xml">
  <output type="rbnb" name="rtmm"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="B200.log" configFile="B200.xml">
  <output type="rbnb" name="B200"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="DC8.log" configFile="DC8.xml">
  <output type="rbnb" name="DC8-5000-cap"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="DIAL.log" configFile="DIAL.xml">
  <output type="rbnb" name="DC8-5001-DIAL-cap"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="CU580.log" configFile="CU580.xml">
  <output type="rbnb" name="CU580"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<httpMonitor initialDirectory="Capture/HttpMonitor" logFile="NOAA_P3.log" configFile="NOAA_P3.xml">
  <output type="rbnb" name="NOAA_P3"/>
</httpMonitor>
<sleep duration_ms="1000"/>
<del initialDirectory="Capture/UDPCapture" file="*.log"/>
<udpCapture initialDirectory="Capture/UDPCapture" logFile="P3B-5700-cap.log">
  <input type="udp" port="5700"/>
  <output type="rbnb" port="localhost:3333" cacheFrames="10" archiveFrames="8640000" overwriteArchive="true" name="P3B-5
</udpCapture>
<udpCapture initialDirectory="Capture/UDPCapture" logFile="P3B-5750-cap.log">
  <input type="udp" port="5750"/>
  <output type="rbnb" port="localhost:3333" cacheFrames="10" archiveFrames="8640000" overwriteArchive="true" name="P3B-5
</udpCapture>
<udpCapture initialDirectory="Capture/UDPCapture" logFile="P3B-5800-cap.log">
  <input type="udp" port="5800"/>
  <output type="rbnb" port="localhost:3333" cacheFrames="10" archiveFrames="8640000" overwriteArchive="true" name="P3B-5
</udpCapture>
<sleep duration_ms="1000"/>
commands started.
Waiting for command completion:
<del initialDirectory="Server" file="rbnb.log"/> complete.
<dataTurbine jvmMaxHeap="1024M" initialDirectory="Server" logFile="rbnb.log" name="INDS" address=":3333" tag="P3B DataTurbine"/>

```

Figure 3. Example of the debug messages displayed by INDS Execution Manager at startup.

Shutdown

Shutdown is a two step process:

1. Shutdown Tomcat

- On Windows: enter Ctrl-C in the Tomcat window
- On Linux/Mac OS X: kill the Tomcat process. Tomcat is a Java process that can be found by executing a command like the following:

```
ps -axww | grep java | grep ClassLoader
```

- Shutdown IEM by entering Ctrl-C in the IEM window. All processes are terminated by IEM in the reverse order from which they were started. Since the IEM *transient* commands (“sleep” and “del”, for instance) are executed and terminate at startup, these commands do not come into play at IEM shutdown. As an example, if IEM starts up RBNB, Sleep, UDPCapture, Sleep, and XMLDemux (in that order), then IEM will terminate commands in the following order: XMLDemux, UDPCapture, and RBNB.

Operation

The *indsViewer* (seen in Figure 1) is comprised of three panels. Each of these, left to right, are discussed in turn below.

Flow Diagram Panel (Left)

The leftmost panel of the *indsViewer* is a flow diagram with clickable icons. The diagram is a schematic that gives an overall dataflow view of the inputs and outputs of a DataTurbine server. The central icon represents the DataTurbine. Data flows from the left-side source icons into the DataTurbine. Data flows from the DataTurbine to the right-side icons of various monitoring applications (sinks) and processing applications (plugins).

Clicking a flow diagram icon causes the associated information about that module to be displayed in the rightmost *Command Information Panel* (discussed below).

Command List Panel (Middle)

The middle *indsViewer* panel is a scrollable list of commands that have been executed by the IEM. Two categories of commands are listed: *current* and *inactive*. Current commands are operational. Inactive commands have completed execution. You can choose to view *Current* or *All* commands via the top-right selector button in this panel.

Clicking a command list panel item causes the associated information about that module to be displayed in the rightmost *Command Information Panel* (discussed below).

Command Information Panel (Right)

The right-side *indsViewer* panel shows information regarding the various INDS modules and commands selected in the left or middle panels.

The top section of the *Command Information Panel* selects the type of information to be displayed in the bottom section of this panel:

Configuration	XML configuration in the IEM for this command.
CommandOut	Standard output of the command as it executes.
CommandError	Error output of the command.
CommandClassification	The type of command: server, source, plugin, converter, transient
ChildConfiguration	The configuration file the command itself uses (if it exists)

Note that the difference between command “standard” and “error” output varies with the command. Check both if you are not sure which output may be relevant for a particular case.

Creating a Project

The INDS Execution Manager distribution includes a sample project, located in *XML_startup_P3_example*, which can be used as a reference/template when creating a new IEM project. We recommend that each IEM project be located in a separate directory at the top level of the IEM hierarchy (that is, alongside the *docs*, *INDSExecutionManager*, *XML_startup_P3_example*, etc. folders). The following files are needed in the project directory:

- IEM “.xml” startup file: The main configuration file for the project. This file specifies all of the commands to be launched by IEM, and the order in which they are to be launched. The name of this file is supplied as an argument when launching IEM. In *XML_startup_P3_example*, this file is called “P3_startup.xml”.
- *inds_startup.xsd*: XML schema file which specifies the format of the IEM “.xml” startup file. This schema is documented in the INDS Execution Manager Reference Manual (located in the *docs* folder).
- *start_tomcat*: Linux/Mac OS X script to launch Tomcat
- *startup.bat*: Windows script to launch Tomcat and IEM. This file must be edited to include the name of the IEM “.xml” startup file as an argument to the IEM command.
- *startup.sh*: Linux/Mac OS X script to launch IEM. This file must be edited to include the name of the IEM “.xml” startup file as an argument to the IEM command.
- *commands.xml*: Specifies information about the various commands that can be launched by IEM. This file will not need to be edited as long as the new project directory is at the top level of the IEM distribution (that is, at the same level as *XML_startup_P3_example*).
- Other supporting directories may be used by IEM, as specified by the project developer in the IEM “.xml” startup file. For example, the P3 directory has various subdirectories: *Capture*, *Jobs*, *Projects*, and *Server*. These subdirectories are used as the run/launch directory for various IEM commands. A directory is associated with a specific command using the XML “initialDirectory” attribute. For example, the following XML snippet from “P3_startup.xml” shows how the *Server* directory is specified as the directory in which the DataTurbine will be executed:

```
<dataTurbine jvmMaxHeap="1024M" initialDirectory="Server" logFile="rbnb.log"
name="INDS" address=":3333" tag="P3B DataTurbine"/>
```

The IEM “.xml” startup file is the main file to be configured for setting up a new project. As mentioned above, the format of this file is specified by a custom schema file, “inds_startup.xsd”. Various tools exist to help author and verify the IEM “.xml” project file against this schema file. The supplied P3 example is a fairly extensive sample which uses all of the basic INDS utilities. This file could be copied to a new project directory and then “whittled down” or adjusted to fit the needs of the new project.

Occasionally, an existing INDS system will be converted to use IEM. This translation task involves taking all of the commands from the existing INDS launch scripts and combining them into the single IEM “.xml” project file. One item to note in this translation task regarding how date format strings are specified for CSVDemux: when launching CSVDemux on the command line or from a script files, the date format string is specified in the following manner:

```
-d yyyyMMdd\'T\'HHmmss
```

However, in the IEM “.xml” project file, the backslashes should not be included. Thus, in the XML file, this would be specified as (note that we use the “dateFormat” attribute in place of the “-d” command line flag):

```
dateFormat="yyyyMMdd\'T\'HHmmss"
```

Reference Documentation

There are two sources of additional reference information, both stored in the Subversion repository (and thus available in your local, IEM folder):

- **Reference Manual:** This manual, written by our collaborators at Creare Inc., contains details on the XML schema, extending the schema (to include a new command, for instance), using the IEM Java RMI interface, and the servlets used along with the IEM. This document can be found in the *docs* directory.
- **xRender Code Documentation:** Doxygen-created documentation for xRender (the Python application which creates the SVG flow diagram), developed by our collaborators at UCSD, can be found at "RBNB\V3.2B1\apache-tomcat-5.5.12\webapps\ROOT\WEB-INF\cgi\docs\html". The "index.html" file is a good starting point for looking at this documentation, as it provides an overview of how the code works as well as access to documentation for all the files.
- **The INDS User Manual and INDS Reference Manual,** supplied to NASA Dryden's GTR under previous work by Creare Inc., provides generally useful information on INDS systems, including an overview of their architecture, operation, and maintenance.

Appendix A: INDS Execution Manager Installation

Installing to a New System

The following steps describe how to install the INDS Execution Manager (IEM) on a new system.

1. Install prerequisites: The following software is required for installing and running IEM.
 - a. *Java*: we recommend at least version 1.5 of the Java Platform, Standard Edition (Java SE); available from <http://java.sun.com/javase/>.
 - b. *Python*: we recommend at least version 2.5.1; available from <http://www.python.org/>.
 - c. *GraphViz*: graph visualization software, used to create the IEM flow chart; available from <http://www.graphviz.org/>.
 - d. *Subversion client*: used to download a copy of the IEM software; available from <http://subversion.tigris.org/>. Windows users may wish to use the TortoiseSVN client, available from <http://tortoisesvn.tigris.org/>.
2. Download the IEM files: IEM is stored in a Google Code Subversion repository at <https://nasa-dryden-dt.googlecode.com/svn/trunk>. The main website for this project on Google Code is <http://code.google.com/p/nasa-dryden-dt>. You obtain a local copy (called a “working copy”) of the IEM files using a Subversion client. First, create a directory called *INDSExecutionManager*, and then go into that directory. Files can be checked-out in one of two ways:
 - a. Checkout a *read-only copy* of the code by executing the following command:

```
svn checkout http://nasa-dryden-dt.googlecode.com/svn/trunk/ nasa-dryden-dt-read-only
```

- b. *Recommended method*: Checkout a *read-write copy* of the code (for example, to join in the development of this project). You must have the following:
 - a general Google account
 - get registered with the nasa-dryden-dt project as a developer
 - have a Google Code password; to get this password, login to Google using your regular Google account and then obtain your password by following the instructions at <http://code.google.com/p/nasa-dryden-dt/source/checkout>.

Once you have followed the above steps, execute the following command (substituting your username and password in the appropriate places); note that “https” is used in this command:

```
svn checkout https://nasa-dryden-dt.googlecode.com/svn/trunk --username <user> --password <pwd> trunk
```

3. RBNB V3.2B3 is included in the repository. If a new version is desired it can be fetched from the Open Source DataTurbine Initiative Google Code project site,

<http://code.google.com/p/dataturbine/>. If a new RBNB is installed, you must follow the instructions at “Updating RBNB and Tomcat for an Existing Installation”, found below.

4. The INDS Utility applications are included in the repository. The official repository for these utility applications is in a Subversion server hosted by NASA Dryden. If the location of the INDS Utilities directory is changed in IEM, you must update the “executableDirectory” entries in commands.xml.
5. To make sure all the most up-to-date servlet files are in place, go to the Tomcat webapps directory (<IEM install dir>/RBNB/V3.2B3/apache-tomcat-6.0.18/webapps) and delete indsExec, indsExec.war, indsViewer, and indsViewer.war. Then copy indsExec.war and indsViewer.war from <IEM install dir>/INDSExManServlet over to this webapps directory.
6. In your project directory (for example, *XML_startup_P3_example*), update the paths in the following files:
 - a. For Windows users: Edit “startup.bat” to include the correct path to Tomcat. Also, Windows users must have JAVA_HOME and CATALINA_OPTS defined in their environment.
 - b. For Mac/Linux users: Edit the CATALINA_HOME and JAVA_HOME variables in “start_tomcat”.
 - c. Update the “dataTurbineDirectory” entry in commands.xml.
7. On Mac/Linux:
 - a. Make sure the “.sh” files located in <IEM install dir>/RBNB/V3.2B3/apache-tomcat-6.0.18/bin have execute permission.
 - b. In the startup directory for the scenario of interest (for example, *XML_startup_P3_example*) make sure “start_tomcat” as well as all of the “.sh” files have execute permission.
8. Edit <IEM install dir>/RBNB/V3.2B3/apache-tomcat-6.0.18/webapps/ROOT/WEB-INF/cgi/defaults.cfg: Set “dotcmd” field to be the full path to the “dot” binary (e.g. /usr/local/bin/dot)

Updating RBNB and Tomcat for an Existing Installation

The as-delivered IEM software (described above) includes a Tomcat and RBNB server software installation already configured to work with the IEM. The following steps describe how to update and configure a new Tomcat/RBNB software package in an existing IEM:

1. Update the definition of “jakartadir” in INDSExManServlet/build.xml to point to the new Tomcat installation. Then, to install indsExec.war and indsViewer.war in the new <Tomcat install>/webapps directory, pop up a console window, change to the INDSExManServlet directory and execute “ant clean” followed by “ant” (this will recompile and reinstall the servlet software).
2. Enable CGI in Tomcat. The specific steps to follow for enabling CGI can depend on the version of Tomcat. As a reference, in Tomcat 6 we added the following <servlet>, <servlet-mapping>, and SVG mappings in apache-tomcat-6.0.18\webapps\ROOT\WEB-INF\web.xml (within the <web-app></web-app> object):

```
<!-- Enable CGI -->
<servlet>
  <servlet-name>cgi</servlet-name>
  <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>6</param-value>
  </init-param>
  <init-param>
    <param-name>executable</param-name>
    <param-value>python</param-value>
  </init-param>
  <init-param>
    <param-name>cgiPathPrefix</param-name>
    <param-value>WEB-INF/cgi</param-value>
  </init-param>
  <init-param>
    <param-name>passShellEnvironment</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>10</load-on-startup>
</servlet>

<!-- The mapping for the CGI Gateway servlet -->
<servlet-mapping>
  <servlet-name>cgi</servlet-name>
  <url-pattern>/cgi-bin/*</url-pattern>
</servlet-mapping>

<!-- JSPC servlet mappings for SVG -->
<mime-mapping>
  <extension>svg</extension>
  <mime-type>image/svg+xml</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>svgz</extension>
  <mime-type>image/svg+xml</mime-type>
</mime-mapping>
```

Also, in `apache-tomcat-6.0.18/conf/context.xml`, you must change `<Context>` to `<Context privileged="true">`.

As a side note, Bill Finger has rightly noted that the above steps make the entire “ROOT” web-app privileged. We enable CGI in ROOT because this is where the Python application resides for constructing the SVG graph. Bill notes that if the Python application were in its own web-app (separate from ROOT) then only that web-app would need to be privileged – this would make the overall system more secure.

3. Copy the contents of the CGI directory (containing Python scripts) from the old IEM installation to the new IEM installation at `apache-tomcat-6.0.18/webapps/ROOT/WEB-INF/cgi`.
4. Create a new, empty directory at `apache-tomcat-6.0.18/webapps/ROOT/inds-svg`
5. Make sure each IEM project uses the new RBNB/Tomcat installation by doing the following in each project directory (for example, *XML_startup_P3_example*):
 - a. Update all of the RBNB and tomat/apache references in `commands.xml`.
 - b. Update the startup `*.bat`, `start_tomcat`, and `*.sh` files to point to the correct RBNB installation.

Appendix B: Verify System Operation/Troubleshooting

The following methods can be used to verify that the INDS Execution Manager is operating correctly:

1. In a browser, go to <http://localhost/cgi-bin/dotFinder.py>. This should return a web page containing the settings used to construct the IEM flow diagram (settings specified in RBNB/V3.2B1/apache-tomcat-5.5.12/webapps/ROOT/WEB-INF/cgi/defaults.cfg).
2. Execute INDSExManGUI (see Figure 4), a Java application which provides much of the same information as the Web-based “indsViewer” interface (but does not require the Tomcat server to be running). To run this application, go to the INDSExManGUI/dist directory and execute:

```
java -jar INDSExManGUI.jar
```

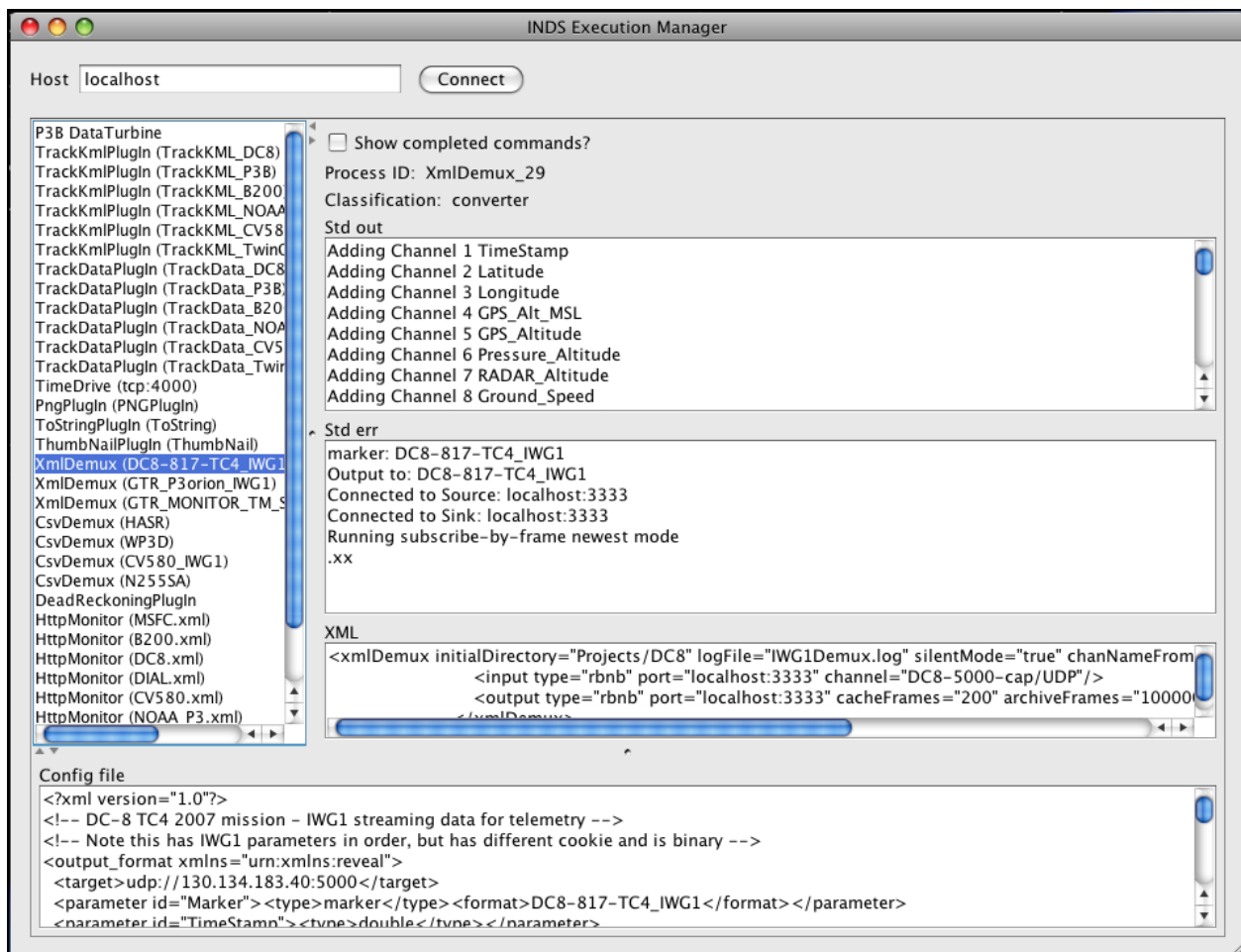


Figure 4. INDSExManGUI

3. Use the "raw" web service commands, for example, for the P3 sample project provided with IEM: http://localhost/indsExec?action=getConfiguration&cmd=CsvDemux_37. See the INDS Execution Manager Reference Manual (in the *docs* directory) for information on the available commands.
4. The IEM data flow chart can be displayed in a web browser by going to:
<http://localhost/cgi-bin/xRender.py>
5. "indsViewer" is the full-featured Web-based IEM interface, available at:
<http://localhost/indsViewer>