Oracle® Fusion Middleware
Administering Oracle Enterprise Data Quality

12*c* (12.2.1.3.0)

E97548-01

July 2018

ORACLE®

Oracle Fusion Middleware Administering Oracle Enterprise Data Quality, 12*c* (12.2.1.3.0)

E97548-01

Contents

1   Using Autorun to Execute Startup Tasks

2   Configuring EDQ Email Notifications

3   Configuring EDQ Case Management

4   Tuning EDQ Performance

5   Using JMX Extensions to Monitor EDQ

6   Using Triggers

7   Accessing EDQ Files Remotely

8   Third-Party License Attributions

Preface

This document describes how to administer and configure Enterprise Data Quality. You can perform a variety of administration tasks to extend the default EDQ configuration.

Audience

This document is intended for system administrators or application developers who are installing the Enterprise Data Quality. It is assumed that you have a basic understanding of core EDQ concepts, application server and web technology and have a general understanding of Linux, UNIX, and Windows platforms.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information about EDQ, see the following documents in the Enterprise Data Quality documentation set.

EDQ Documentation Library

The following publications are provided to help you install and use EDQ:

- *Release Notes for Enterprise Data Quality*
- *Installing and Configuring Enterprise Data Quality*
- *Administering Enterprise Data Quality*
- *Understanding Enterprise Data Quality*
- *Integrating Enterprise Data Quality With External Systems*
- *Securing Oracle Enterprise Data Quality*
- *Installation and Upgrade Guide*
- *Release Notes*

Find the latest version of these guides and all of the Oracle product documentation at

https://docs.oracle.com

Online Help

Online help is provided for all Enterprise Data Quality user applications. It is accessed in each application by pressing the **F1** key or by clicking the Help icons. The main nodes in the Director project browser have integrated links to help pages. To access them, either select a node and then press **F1**, or right-click on an object in the Project Browser and then select **Help**. The EDQ processors in the Director Tool Palette have integrated help topics, as well. To access them, right-click on a processor on the canvas and then select **Processor Help**, or left-click on a processor on the canvas or tool palette and then press **F1**.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Using Autorun to Execute Startup Tasks

This chapter provides an introduction to the EDQ autorun functionality, which allows EDQ to load projects and run jobs when the application server starts up. It explains how the autorun functionality is configured, introduces the chore types that can be performed by using the autorun facility and provides examples of autorun scripts. This chapter includes the following sections:

- Understanding Autorun
- Using the Autorun Chores
- Using the Autorun Scripts
- Understanding the Chore and Rules Schemas

Understanding Autorun

EDQ can be configured to do the following automatically at startup:

- Perform a range of tasks when the application server starts up. Each task, which is composed of chores, can be configured to be performed every time the application server is started, or just once the next time the application server is started.
- Load and apply purge rules that override the purge settings that are stored in the EDQ server.

To use autorun processing, you place autorun scripts, written in XML, that specify tasks in one of two specific directories in the EDQ installation:

- `startup` directory: Scripts in the `startup` directory are processed every time the EDQ application server starts up.
- `onceonly` directory: Scripts in the `onceonly` directory are processed when the EDQ application server next starts up, and are then moved to the `complete` subdirectory within `onceonly`. Scripts in the `complete` directory are not processed on subsequent start ups.

When the application server starts up, EDQ checks the `onceonly` and `startup` directories for autorun scripts and processes any that are present.

The `startup` and `onceonly` directories are located in the EDQ `autorun` directory in the local configuration directory of the application server, `oedq.local.home`.

Using the Autorun Chores

Various kinds of autorun chores are available in EDQ, each with a set of XML attributes specific to its function. The chore types and their available attributes are defined by the autorun file XML schema, see Understanding the Chore and Rules Schemas. The chores available are listed in the following table:

Chore Type	What the Chore Does
`httpget`	Downloads files from a web server.
`package`	Loads a project from a `.dxi` file into the server, or saves a project on the server into a `.dxi` file. If no nodes are specified then the contents of the whole file, including system level components, are loaded into the server.
`load`	Loads a file, for example a purge rules configuration file. This chore is valid only in the `startup` directory. See Example 3, #unique_19/ unique_19_Connect_42_BABCGHFA for how to use the `load` chore with the `Rules` schema to load purge rules.
`runjob`	Runs an existing job from Director. Any run labels in a run profile specified in this chore are ignored. (Use `runopsjob` to run a job based on a run label.)
`runopsjob`	Runs an existing job from the EDQ Server Console and requires a run label to be set, either in the run profile or with the `runlabel` attribute.
`dbscript`	Runs a database script against the Director database. This kind of chore must only be used with extreme care, as inappropriately applied scripts may corrupt the underlying database.
`sleep`	Waits for a specified interval before proceeding.

Using the Autorun Scripts

Autorun scripts are files that contain XML code. The main part of an autorun script consists of a list of chores, each bounded by `<chores>` tags. Each chore is of one of the autorun chore types listed in Using the Autorun Chores and includes a set of attributes that specify the chore to be performed. The attributes available depend on the chore type selected.

The XML schema that is used to structure autorun scripts is shown in full in Understanding the Chore and Rules Schemas.

Examples

This section shows some examples of autorun scripts.

Example 1
The following XML code shows a sample autorun script that instructs EDQ to:

- Download the `23People.dxi` file, overwriting any existing file with the same name.

- Import the `23People` project from the `23People.dxi` file, overwriting any existing project with the same name.

- Run the `23People Excel.23People` job with the `rp1` run profile. Any run label specified in the profile will be ignored, because this is not a `runopsjob` chore.

```
<?xml version="1.0" encoding="UTF-8"?>
 <chores version="1">
  <!-- Get the dxi file -->
    <httpget overwrite="true" todir="dxiland" tofile="23People.dxi">
      <url>http://svn/repos/dev/trunk/benchmark/ benchmark/dxis/23People.dxi</url>
  </httpget>
  <!-- Import the project from the dxi -->
    <package direction="in" dir="dxiland" file="23People.dxi" overwrite="true">
```

```
        <node type="project" name="23People"/>
    </package>
    <!-- Run the jobs -->
    <runjob project="23People" job="23People Excel.23People" runprofile="rp1"
      waitforcompletion="true"/>
 </chores>
```

Example 2

The following XML code shows a sample autorun script that shows four different ways to use a `runjob` or `runopsjob` chore to run a job.

```
<?xml version="1.0" encoding="UTF-8"?>
<chores version="1">
  <!-- runs a director job with no runlabel -->
  <runjob project="merge" job="tester" waitforlocks="false"
    waitforcompletion="false" runprofile="x"/>
  <!-- runs an ops job with the runlabel from the runprofile -->
  <runopsjob project="merge" job="tester" waitforlocks="false"
    waitforcompletion="false" runprofile="x" />
  <!-- runs an ops job with the runlabel from the runlabel attribute-->
  <runopsjob project="merge" job="tester" waitforlocks="false"
     waitforcompletion="false" runprofile="x" runlabel="chooseme" />
  <!-- runs an ops job with the runlabel from the runlabel attribute-->
  <runopsjob project="merge" job="tester" waitforlocks="false"
     waitforcompletion="false" runlabel="onlychoice" />
</chores>
```

Example 3

The following XML code shows how to use a load chore to load purge rules.

```
<?xml version="1.0" encoding="UTF-8" ?>
<chores version="1">
  <load file="purgerules.xml" dir="autorun" type="purgeRules" />
</chores>
```

The following are the purge rules in the `purgerules.xml` file that is loaded in the chore specification:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <rules>
  - <rule displayName="testa" enabled="true">
      <purgePeriod period="1" unit="HOURS" />
      <project>aa</project>
      <job>12345</job>
      <runlabelMatcher regex="false" runlabel="ABCD" />
  </rule>
  - <rule displayName="testb" enabled="true">
      <purgePeriod period="1" unit="HOURS" />
      <project>aa</project>
      <job>ABCD</job>
      <runlabelMatcher regex="true" runlabel="^\d{5}$" />
  </rule>
- <rule displayName="testc" enabled="true">
  <purgePeriod period="2" unit="HOURS" />
      <project />
      <job />
      <runlabelMatcher regex="true" runlabel="TEST" />
  </rule>
- <rule displayName="testd" enabled="true">
```

```
                            <purgePeriod period="3" unit="WEEKS" />
                            <project />
                            <job />
                            <runlabelMatcher regex="true" runlabel="TEST" />
                      </rule>
                      - <rule displayName="teste" enabled="false">
                            <purgePeriod period="999" unit="MONTHS" />
                            <project />
                            <job />
                            <runlabelMatcher regex="true" runlabel="^\d{5}$" />
                      </rule>
                      - <rule displayName="testf" enabled="true">
                            <purgePeriod period="1" unit="HOURS" />
                            <project />
                            <job />
                            <runlabelMatcher regex="true" runlabel="^\d{5}$" />
                      </rule>
                      - <rule displayName="testg" enabled="true">
                            <purgePeriod period="1" unit="DAYS" />
                            <project />
                            <job />
                            <runlabelMatcher regex="false" runlabel="ABCD" />
                      </rule>
                </rules>
```

Understanding the Chore and Rules Schemas

This section shows the Chores and Rules XML schemas.

Understanding the Chores Schema

This schema explains the chores listed in Using the Autorun Chores.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Chores -->
  <xs:element name="chores">
    <xs:complexType>

      <!--
      List of chores that need to be performed.  The chores will be performed
      in the order
      specified in the xml file
      -->
      <xs:choice minOccurs="0" maxOccurs="unbounded">

        <xs:element name="httpget"     type="httpgetType"/>
        <xs:element name="package"     type="packageType"/>
        <xs:element name="runjob"      type="runjobType"/>
        <xs:element name="runopsjob"   type="runopsjobType"/>
        <xs:element name="dumpdb"      type="dumpdbType"/>
        <xs:element name="dbscript"    type="dbScriptType"/>
        <xs:element name="sleep"       type="sleepType"/>
        <xs:element name="load"        type="loadType"/>
      </xs:choice>

      <!-- Schema version number -->
      <xs:attribute name="version" type="xs:positiveInteger" use="required"/>
```

```
        </xs:complexType>
</xs:element>

<!-- Base type for chores -->
<xs:complexType name="choreType">

  <!-- Flag indicating whether we should wait for completion before moving
  on to the next chore. -->
  <xs:attribute name="waitforcompletion" type="xs:boolean"
  use="optional" default="true"/>
</xs:complexType>

<!-- HTTP Get chore.  Download the specified urls. -->
<xs:complexType name="httpgetType">

  <xs:complexContent>
    <xs:extension base="choreType">

      <xs:sequence minOccurs="1" maxOccurs="1">
        <!-- URL to download. -->
        <xs:element name="url" type="xs:string"/>
      </xs:sequence>

      <!-- Filename to download to. -->
      <xs:attribute name="tofile" type="xs:string" use="required"/>

      <!--
      Directory to download the files to.
        - relative path is relative to the config dir
        - absolute path is used as is
        - no path indicates the config dir
      -->
      <xs:attribute name="todir" type="xs:string" use="optional"/>

      <!-- If true existing files are overwritten, otherwise download is
      not performed. -->
      <xs:attribute name="overwrite" type="xs:boolean" use="optional"
      default="true"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- dxi file control chore.  Import or export to/from a dxi file. -->
<xs:complexType name="packageType">

  <xs:complexContent>
    <xs:extension base="choreType">

      <!-- List of root level nodes to import/export.
      An empty list indicates 'all'. -->
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="node" type="packageNodeType"/>
      </xs:sequence>

      <!-- dxi filename. -->
      <xs:attribute name="file" type="xs:string" use="required"/>

      <!--
      Directory that the dxi is in.
        - relative path is relative to the config dir
```

```
        - absolute path is used as is
        - no path indicates the config dir
      -->
      <xs:attribute name="dir" type="xs:string" use="optional"/>

      <!-- If true existing files/nodes are overwritten,
      otherwise no operation. -->
      <xs:attribute name="overwrite" type="xs:boolean"
      use="optional" default="true"/>

      <!-- Direction: in=import out=export -->
      <xs:attribute name="direction" type="packageDirectionEnum"
      use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Package node for import or export from/to a dxi. -->
<xs:complexType name="packageNodeType">

  <!-- the type of the node to process -->
  <xs:attribute name="type" type="nodeTypeEnum" use="required"/>

  <!-- the name of the node to process -->
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>

<!-- db script control chore.  Runs db script against the configuration database.
-->
<xs:complexType name="dbScriptType">

  <xs:complexContent>
    <xs:extension base="choreType">

      <!-- db script filename. -->
      <xs:attribute name="file" type="xs:string" use="required"/>

      <!--
      Directory that the db script is in.
        - relative path is relative to the config dir
        - absolute path is used as is
        - no path indicates the config dir
      -->
      <xs:attribute name="dir" type="xs:string" use="optional"/>

      <!-- The database to run the script against -->
      <xs:attribute name="database" type="databaseEnum" use="required"/>

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Invoke named job chore.  Run a named job -->
<xs:complexType name="runjobType">
  <xs:complexContent>
    <xs:extension base="choreType">

      <!-- Project name -->
      <xs:attribute name="project" type="xs:string" use="required"/>

      <!-- Job name -->
```

```
                <xs:attribute name="job" type="xs:string" use="required"/>

                <!-- Wait for locks flag - default to true -->
                <xs:attribute name="waitforlocks" type="xs:boolean"
                use="optional" default="true"/>

                <!-- Optional run profile -->
                <xs:attribute name="runprofile" type="xs:string" use="optional"/>

            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="runopsjobType">
        <xs:complexContent>
            <xs:extension base="runjobType">

                <!-- Optional run label (will override run profile run label if set) -->
                <xs:attribute name="runlabel" type="xs:string" use="optional"/>

            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!--
      Dump the database.
    -->
    <xs:complexType name="dumpdbType">
        <xs:complexContent>
            <xs:extension base="choreType">

                <!-- Output JMP file for config database -->
                <xs:attribute name="configout" type="xs:string" use="required"/>

                <!-- Output JMP file for results database -->
                <xs:attribute name="resultsout" type="xs:string" use="required"/>

                <!--
                Directory that the JMP files are written to
                  - relative path is relative to the config dir
                  - absolute path is used as is
                  - no path indicates the config dir
                -->
                <xs:attribute name="dir" type="xs:string" use="optional"/>

                <!--
                TODO: Add some filtering to allow dumping of categories of data
                e.g. staged data, results data, case management data, etc.
                -->
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!-- Load a certain file to do a certain thing. Eg change purge rules. -->
    <xs:complexType name="loadType">
        <xs:complexContent>
            <xs:extension base="choreType">
                <!-- type of action to run with file -->
                <xs:attribute name="type" type="loadTypeEnum" use="required"/>

                <!-- filename -->
```

```
        <xs:attribute name="file" type="xs:string" use="required"/>

        <!--
        Directory that the file is in.
          - relative path is relative to the config dir
          - absolute path is used as is
          - no path indicates the config dir
        -->
        <xs:attribute name="dir" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <!-- Enumeration of databases  -->
  <xs:simpleType name="databaseEnum">
    <xs:restriction base="xs:string">
        <xs:enumeration value="director"/>
        <xs:enumeration value="results"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- Enumeration of valid node types -->
  <xs:simpleType name="nodeTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="project"/>
      <!-- Probably need to do these sometime
      <xs:enumeration value="resource"/>
      <xs:enumeration value="datastore"/>
      -->
    </xs:restriction>
  </xs:simpleType>

  <!-- Enumeration of packaging direction. -->
  <xs:simpleType name="packageDirectionEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="in"/>
      <xs:enumeration value="out"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- Enumeration of types of things that can be loaded. -->
  <xs:simpleType name="loadTypeEnum">
    <xs:restriction base="xs:string">
      <xs:enumeration value="purgeRules"/>
      <!-- <xs:enumeration value="schedule"/> -->
    </xs:restriction>
  </xs:simpleType>

  <!-- Sleep chore.  Wait for a while before doing other autorun stuff -->
  <xs:complexType name="sleepType">

    <xs:complexContent>
      <xs:extension base="choreType">

        <!-- seconds to wait. -->
        <xs:attribute name="time" type="xs:integer" use="required"/>

      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```
                    </xs:schema>
```

Understanding the Rules Schema

This section describes the Rules schema, which provides the basis for structuring an XML script that specifies EDQ server purge rules. Use the `load` chore to load the script at EDQ startup.

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Common types -->
  <!-- ============ -->
  <xs:include schemaLocation="urn:commontypes.xsd"/>

  <xs:element name="rules" type="rulesType">
    <!-- Rule name must be unique -->
    <xs:key name="rule.name">
      <xs:selector xpath="rules/rule"/>
      <xs:field    xpath="@name"/>
    </xs:key>
  </xs:element>

  <!--  Rules  -->

  <xs:complexType name="rulesType">
    <xs:sequence>
      <xs:element name="rule" type="ruleType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>

    <xs:attribute name="schemaversion" type="xs:positiveInteger"
      use="optional" default="1"/>
  </xs:complexType>

  <xs:complexType name="ruleType">
    <xs:sequence>
      <xs:element name="purgePeriod"     type="periodType"   minOccurs="1"
        maxOccurs="1"/>
      <xs:element name="project"         type="xs:string"    minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="job"             type="xs:string"    minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="runlabelMatcher" type="runlabelType" minOccurs="0"
        maxOccurs="1"/>
    </xs:sequence>

    <!-- name -->
    <xs:attribute name="displayName"   type="xs:string"  use="required"/>
    <!-- whether this rule should be applied -->
    <xs:attribute name="enabled" type="xs:boolean" use="required"/>
  </xs:complexType>

  <!-- Runlabel -->

  <xs:complexType name="runlabelType">
    <xs:attribute name="regex"    type="xs:boolean" use="required"/>
    <xs:attribute name="runlabel" type="xs:string"  use="required"/>
```

```
        </xs:complexType>

        <!-- Purge Period -->

        <xs:complexType name="periodType">
          <xs:attribute name="period" type="xs:int" use="optional"/>
          <xs:attribute name="unit" type="periodUnitType" use="required"/>
        </xs:complexType>

        <!-- Purge Unit types -->

        <xs:simpleType name="periodUnitType">
          <xs:restriction base="xs:string">
            <xs:enumeration value="IMMEDIATE"/>
            <xs:enumeration value="HOURS"/>
            <xs:enumeration value="DAYS"/>
            <xs:enumeration value="WEEKS"/>
            <xs:enumeration value="MONTHS"/>
            <xs:enumeration value="NEVER"/>
          </xs:restriction>
        </xs:simpleType>

    </xs:schema>
```

2

Configuring EDQ Email Notifications

This chapter describes how to configure EDQ to produce email notifications in a number of situations.

- Using SMTP to Send Email Notifications
- Configuring Email Sessions in WebLogic Administration Console
- Ensuring that Email is Configured

Emails can be sent to EDQ users when relevant issues are created or changed, when relevant cases or alerts in Case Management are added or modified, or when relevant jobs are finished running.

Using SMTP to Send Email Notifications

To send email notifications, the Simple Mail Transfer Protocol (SMTP) information for your EDQ installation must be entered in the `mail.properties` file. Email authentication from EDQ works with an SMTP server requiring authentication is now supported via WebLogic credentials store. Set the SMTP user name and password as the session user name and password to the property list. This `mail.properties` file is stored in `/oedq_home/notification/smtp`.

1. Copy the `mail.properties` file from its installed location of `edq_home/notification/smtp` to the `notification/smtp` sub-directory of the local configuration directory (`oedq_local_home` by default).

   `/oedq_local_home/notification/smtp`

   This file is in the standard Java `mail.properties` file format, as documented at the JavaMail API documentation website found at `https://javamail.java.net/nonav/docs/api/`.

2. Edit the `mail.properties` file as follows, supplying the name of your SMTP host at the site.

   ```
   enabled = true
   mail.transport.protocol = smtp
   mail.host = smtp.fully qualified domain name of mail host
   mail.user = depends on client site
   mail.password = depends on client site
   from.address = edqserver@example.com
   ```

Configuring Email Sessions in WebLogic Administration Console

You can also send email notifications by creating the configuration for an email session in the WebLogic console. You can then refer to this by the JNDI name in the mail.properties file. To configure a JNDI-accessible session, see http://docs.oracle.com/middleware/1221/wls/WLACH/taskhelp/mail/CreateMailSessions.html.

```
session = JNDI name of session
from.address = edqserver@example.com
enabled = true
```

> **Note:**
>
> For email notifications to work correctly, you must ensure that the `from.address` property is set to a valid email format for your site. You must also ensure that each of your users who will be receiving email notifications has an email address configured in their profile.

Ensuring that Email is Configured

To check that email notifications are working correctly, create a test issue in Director and assign it to a user with a configured email address. The user should receive an email with a link to the issue.

3

Configuring EDQ Case Management

This chapter describes how to configure EDQ to use Case Management. This chapter includes the following sections:

- Understanding and Adding Extended Attributes
- Configuring Data Entry Validation
- Understanding Case Management Configuration Properties

Case Management supports the manual investigation of results from data quality processes. Using Case Management, privileged users can manage and review matching results using highly configurable workflows.

The complete set of Case Management extended attributes that are used on an EDQ server are configured in the `flags.xml` file in the `oedq_local_home/casemanagement` directory. This file must be modified to add new extended attributes, and to define rules for how these attributes are populated.

An additional property file named `flags.properties` accompanies the base `flags.xml` file and specifies the labels for the extended attributes as they will appear in the graphical user interface (GUI). The settings in this file may be overridden for a specific client language by the creation of additional property files with an ISO 639-1 language code, such as `flags_en.properties` (for English) or `flags_de.properties` (for German). This language code is described at the ISO website found at http://www.iso.org/iso/home/standards/language_codes.htm.

If Oracle Watchlist Screening is installed, these files may already exist.

To ensure that Case Management publication works correctly, the `flags.xml` file is overwritten whenever a Case Source is imported using the Case Management Administration application. This is because Case Sources have a dependency on the format of the `flags.xml` file and requires the flags to be indexed and specified in the same way as on the server where the Case Source was defined. Oracle recommends that you back up the file before importing a Case Source in case there are any existing extended attributes in the `flags.xml` file on the server that need to be re-added once the import is complete.

Understanding and Adding Extended Attributes

This section describes the different types of extended attributes and how to add them for use in Case Management.

Default Extended Attributes

In an initial EDQ installation, the `flags.xml` file contains the following two extended attribute (`flag`) example definitions:

```
<f:flag index="1" label="%escalation" type="boolean" default="false"
notnull="true"/>
```

```
<f:flag index="2" label="%priority.score" type="number" readonly="true"/>
```

> **✎ Note:**
>
> The order in which these properties appear in each line may not match this example. The order of properties is immaterial. Also, if Oracle Watchlist Screening is installed, the contents of the `flags.xml` file is different.

Adding New Extended Attributes

To add a new extended attribute, add a line immediately after the existing attribute definitions in the `flags.xml` file, following the same syntax as the existing lines and using the following notes for each property:

Property	Allowed Values	Notes
index	Integer	Must be unique for each entry in the file
label	Any	The% character is used to indicate that the label for the UI should be retrieved from the `flags.properties` file for the client locale. If the% character is not used, the label will always be exactly as stated (in all languages).
type	number, boolean, or string	Controls the data type of the column.
readonly	true or false	Controls whether or not privileged users can edit the value of the extended attribute when editing a Case or Alert
notnull	true or false	Controls whether or not Null values are allowed in the extended attribute. If this is undefined, Null values are allowed (the same as the 'false' setting).
default	Any permissible value	Sets the default value of the extended attribute if not set to a specific value.

There is a character limit of 80 characters for extended attributes with a type of 'string'. Values longer than this cannot be inserted as values.

Configuring Data Entry Validation

You can restrict the format of user-specified data for an extended attribute. The restriction is checked when users edit extended attributes in the Case Management GUI, and when defining possible values to set for an extended attribute in the Workflow editor in Case Management Administration.

The restriction is not checked when cases and alerts are written to Case Management from a process, so it is possible to write invalid values into an extended attribute. The invalid values will appear in error, with an appropriate error message. This designed behavior protects the system against unnecessary job failure.

Restrictions are defined as part of the `flags.xml` file. There are two types of possible restrictions:

- **Predefined list** means that the data to be written is checked against a predefined list of allowed values.
- **Regular expression** means that the data to be written is checked against a regular expression.

Checking Predefined List Restrictions

To check that the data being entered into the extended attribute matches a predefined list of possible values, add XML elements in the following format after the definition of the extended attribute (`flag`):

```
<f:restrictions>
<f:predefined>
<f:value>first value</f:value>
<f:value>second value</f:value>
<f:value>third value</f:value>
</f:predefined>
</f:restrictions>
</f:flag>
```

For example, the following XML fragment defines a custom 'Status' extended attribute that allows only the values 'active' and 'inactive':

```
<f:flag index="6" label="Status" type="string" readonly="false">
<f:restrictions>
<f:predefined>
<f:value>active</f:value>
<f:value>inactive</f:value>
</f:predefined>
</f:restrictions>
</f:flag>
```

The extended attribute appears with a list of the valid values in the Case Management Edit Case (or Edit Alert) dialog:
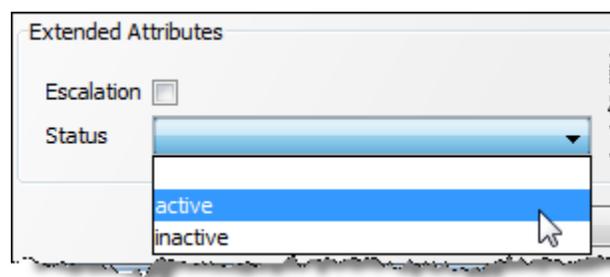
> 💡 **Tip:**
>
> In this case, the user can specify a Null value for the `Status` field (as a 'notnull' condition was not set).

Checking Regular Expression Restriction

To check that data being entered into the extended attribute matches a regular expression, add XML elements in the following format after the definition of the extended attribute (`flag`):

```
<f:restrictions>
<f:regex ignorecase="false" matchby="w">
<f:value></f:value>
</f:regex>
</f:restrictions>
```

Where: the `value` property defines the regular expression, and the `ignorecase` and `matchby` properties defines how it is matched. The possible values for the `matchby` condition are as follows:

Value	Description
w	WHOLE - The whole value must match the Regular Expression.
s	STARTS - The beginning of the value must match the Regular Expression.
e	ENDS - The end of the value must match the Regular Expression.
c	CONTAINS - The value must contain a string that matches the Regular Expression.

For example, the following XML fragment defines a custom 'National ID' extended attribute that allows only values in the format NN-NN-NNN (2 digits, hyphen, 2 digits, hyphen, 3 digits):

```
<f:flag index="7" label="National ID" type="string" readonly="false"
notnull="true">
<f:restrictions>
<f:regex ignorecase="false" matchby="w">
<f:value>\d{2}-\d{2}-\d{3}</f:value>
</f:regex>
</f:restrictions>
</f:flag>
```

The following shows the error message displayed when a user attempts to add a value that does not match the regular expression:
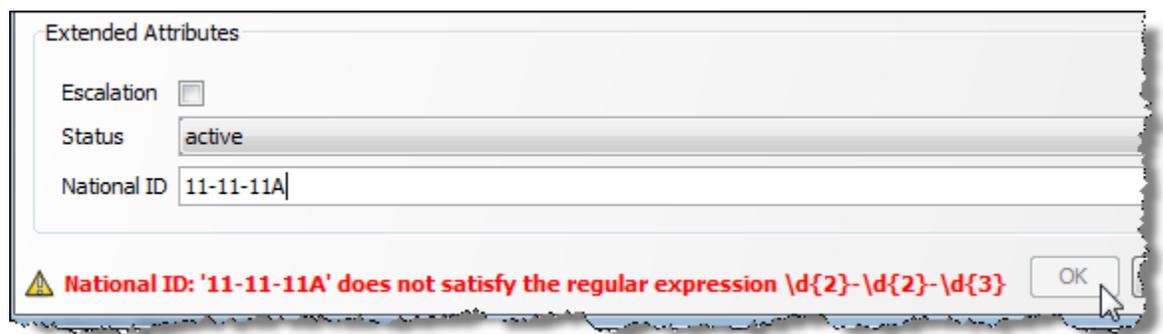
It is also possible to customize this error message with the `errormessage` attribute. Either enter a simple text string to be displayed as the error message, or begin the

string with a percent (%) symbol to direct the application to look in the `flags.properties` file for a localized value.

For example, the following XML fragment causes the `e1.message` error message to be retrieved from the `flags.properties` file when an error occurs:

```
<f:restrictions><f:regex ignorecase="false" matchby="w"
errormessage="%e1.message"><f:value>\d{3}-\d{2}-\d{4}</f:value></f:regex></
f:restrictions>
```

Understanding Case Management Configuration Properties

This section lists the main parameters in `director.properties` that are used to configure Case Management.

Parameter	Description0
`case.management.fail.on.long.flags`	This property controls the Case Management behavior when flag values that are longer than 80 characters are generated. If this property is set to `true`, the process will generate an error and will stop. If it is set to `false`, long flag values will be truncated and a warning will be written to the log file. This property is set to `false` by default.
`cm.index.queue.limit`	This property controls the maximum size of the index queue limit.
`index.directory`	This property allows an absolute path for the Lucene index directories to be configured. By default, the index directories are always created within the `localhome` directory. In some circumstances, these directories can become very large, and storing them in a separate location may facilitate better management of disk space.

4

Tuning EDQ Performance

This chapter describes the server properties that can be used to optimize the performance of the EDQ system and how these properties should be configured in various circumstances.
This chapter includes the following topics:

- Understanding the Properties File

- Tuning for Batch Processing

- Tuning for Real-Time Processing

- Tuning JVM Parameters

- Tuning Database Parameters

- Adjusting the Client Heap Size

EDQ has a large number of properties that are used to configure various aspects of the system. A relatively small number of these are used to control the performance characteristics of the system.

Performance tuning in EDQ is often discussed in terms of **CPU cores**. In this chapter, this refers to the number of CPUs reported by the Java Virtual Machine as returned by a call to the `Runtime.availableProcessors()` method.

Understanding the Properties File

The tuning controls are exposed as properties in the `director.properties` file. This file is found in the `oedq_local_home` configuration directory.

The available tuning properties are as follows:

Properties	Description
`runtime.threads`	This property determines the number of threads that will be used for each batch job which is invoked. The default value of this property is zero, meaning that the system should start one thread for each CPU core that is available. You can specify an explicit number of threads by supplying a positive, non-zero integer as the value of this property. For example, if you know that you want to start a total of four threads for each batch process, set `runtime.threads` to four.
`runtime.intervalthreads`	This property determines the number of threads that will be used by each process when running in interval mode. This will also define the number of requests that can be processed simultaneously. The default behavior is to run a single thread for each process running in interval mode.

Properties	Description
`workunitexecutor.outputThreads`	This property determines the number of threads that will be used to write data to the results database. These threads service the queue of results and output data for the whole system, and so are shared by all the processes which are running on the system. The default value of this property is zero, meaning that the system should use one output thread for each CPU core that is available. You can specify an explicit number of output threads by supplying a positive, non-zero integer as the value of this property. For example, if you know that you want to use a total of four threads for each batch process, set `workunitexecutor.outputThreads` to 4.

Tuning for Batch Processing

The default tuning settings provided with EDQ are appropriate for most systems that are primarily used for batch processing. Enough threads are started when running a job to use all available cores. If multiple jobs are started, the operating system can schedule the work for efficient sharing between the cores. It is best practice to allow the operating system to perform the scheduling of these kinds of workloads.

Tuning for Real-Time Processing

When a production system is being used for a significant amount of real time processing, it should not be used for simultaneous batch and real time processing unless the real time response is not critical. Run batch processing only to process data that is required by the real time processes.

Tuning Batch Processing On Real-Time Systems

If batch processing must be run on a system that is being used for real time processing, it is best practice to run the batch work when the real time processes are stopped, such as during a scheduled maintenance window. In this case, the default setting of `runtime.threads` is appropriate.

If it is necessary to run batch processing while real time services are running, set `runtime.threads` to a value that is less than the total number of cores. By reducing the number of threads started for the batch processes, you prevent those processes from placing a load on all of the available cores when they run. Real time service requests that arrive when the batch is running will not be competing with it for CPU time.

Tuning Real-Time Thread Numbers

For most production systems the default value of one for `runtime.intervalthreads` is not appropriate. The default setting implies that, for any given real-time service handled by a process running in interval mode, all requests will be processed sequentially. If four requests for the same service arrive simultaneously, and the average time to process a request is 100 ms, then the first message will be processed after 100 ms, the second after 200 ms, and so on. In addition, all the work will be performed by a single core, meaning that on a four-core machine three of the cores are idle. It is best practice to set `runtime.intervalthreads` to the same as the number of available cores. This configuration allows incoming requests to be processed simultaneously, resulting in a more efficient use of resources and a much faster

turnaround speed. The default setting for `runtime.intervalthreads` is adequate for development environments.

Tuning I/O Heavy Real-Time Processes

If a process performs significant I/O, you can try increasing the value of `runtime.intervalthreads` above the number of available cores. When a process performs intensive I/O, there will be times when all the threads are waiting for disk activity to complete, leaving one or more cores idle. By using more active threads than there are cores, you ensure that when one thread stalls for I/O, another thread can utilize the core that the thread was using.

Example of Tuning Real-Time Processes

In this example of how to tune real-time processes, a four-core Intel server is being used to support four different web services. The web services are CPU-intensive and perform minimal amounts of I/O. Some data used by the web services must be updated on a daily basis, which includes running a data preparation process in a batch mode. The web services receive intermittent sets of simultaneous requests. Overnight, the web services are stopped for maintenance and data preparation.

In this scenario, it is appropriate to leave the `runtime.threads` property set to its default value of one thread per CPU core: in this case, four threads. With the goal of performing data preparation in the quickest possible time, and assuming the process is not likely to become I/O bound, you can set the `runtime.intervalthreads` property to four. Using the same number of threads as processes ensures that the maximum number of requests are processed at the same time.

> **Note:**
>
> Increasing the value of `runtime.intervalthreads` means that there will be a significant increase in the memory requirement, particularly at interval turnover.

Tuning JVM Parameters

JVM parameters should be configured during the installation of EDQ. For more information, see *Installing and Configuring Enterprise Data Quality*. If it becomes necessary to tune these parameters post-installation to improve performance, follow the instructions in this section.

> **Note:**
>
> All of the recommendations in this section are based on EDQ installations using the Java HotSpot Virtual Machine. Depending on the nature of the implementations, these recommendations may also apply to other JVMs.

Setting the PermGen Space

If the following error message is reported in the log file, it may be necessary to increase the maximum PermGen space available:

```
java.lang.OutOfMemoryError: PermGen space
```

To do this, change the value against the `-XX:MaxPermSize` parameter on the JVM on the EDQ server. It will also be necessary to change the `-XX:ReservedCodeCacheSize` parameter proportionally. For example, if the `MaxPermSize` is doubled from 1024m to 2048m, the `ReservedCodeCacheSize` should be doubled.

Setting the Maximum Heap Memory

If an `OutOfMemory` error message is generated in the log file, it may be necessary to increase the maximum heap space parameter, `-Xmx`. For most use cases, a setting of 8GB is sufficient. However, large EDQ installations may require a higher max heap size, and therefore setting the `-Xmx` parameter to a value half that of the server memory is the normal recommendation.

Tuning Database Parameters

The most significant database tuning parameter with respect to performance tuning within EDQ is `workunitexecutor.outputThreads`. This parameter determines the number of threads, and hence the number of database connections, that will be used to write results and staged data to the database. All processes that are running on the application server share this pool of threads, so there is a risk of processing becoming I/O bound in some circumstances. If there are processes that are particularly I/O intensive relative to their CPU usage, and the database machine is more powerful than the machine hosting the EDQ application server, it may be worth increasing the value of `workunitexecutor.outputThreads`. The additional database threads would use more connections to the database and put more load on the database.

Adjusting the Client Heap Size

Under certain conditions, client heap size issues can occur; for example, when:

- attempting to export a large amount of data to a client-side Excel file, or

- opening up Match Review when there are many groups.

EDQ allows the client heap size to be adjusted using a property in the `blueprints.properties` file.

To double the default maximum client heap space for *all* Java Web Start client applications, create (or edit if it exists) the file `blueprints.properties` in the *local* configuration directory of the EDQ server. For more information about the EDQ configuration directories, see "EDQ Directory Requirements" in *Installing Oracle Enterprise Data Quality*.

Add the line:

```
*.jvm.memory = 512m
```

> **✎ Note:**
>
> Increasing this value will cause all connecting clients to change their heap sizes to 512MB. This could have a corresponding impact on client performance if other applications are in use.

To adjust the heap size for a specific application, replace the asterisk, `*`, with the blueprint name of the client application from the following list:

- `director` - (Director)
- `matchreviewoverview` - (Match Review)
- `casemanager` - (Case Management)
- `casemanageradmin` - (Case Management Administration)
- `opsui` - (Server Console)
- `diff` - (Configuration Analysis)
- `issues` - (Issue Manager)

> **✎ Note:**
>
> Dashboard is not a Java Web Start application, and therefore cannot be controlled using this property.

For example, to double the maximum client heap space for Director, add the following line:

```
director.jvm.memory = 512m
```

When doubling the client heap space for more than one application, simply repeat the property; for example, for Director and Match Review:

```
director.jvm.memory = 512m

matchreviewoverview.jvm.memory = 512m
```

5

Using JMX Extensions to Monitor EDQ

This chapter describes theEDQ Java Management Extensions (JMX) interface that can be used to monitor and manage many details of its operation. JMX is a Java technology designed for remote administration and monitoring of Java components This chapter includes the following topics:

- Understanding JMX Binding
- Understanding JMX Bean Naming
- Monitoring Real-Time Processes

Understanding JMX Binding

EDQ can use either an internal JMX server or one that is provided in the WebLogic or Tomcat application server. This topic explains how to control which JMX server is used.

- A default installation of EDQ on Apache Tomcat uses an internal JMX server.
- A default installation of EDQ on Oracle WebLogic Server uses the JMX tree in the WebLogic Server application server.

The default configuration contains a Remote Method Invocation (RMI) registry, which is used by the EDQ command line interface as well as by JMX clients. The RMI listening port number is specified by the `management.port` property, defined in the `director.properties` file. The default is 8090. This property controls access to both the internal JMX Server and the RMI API that is used by the EDQ command line tools.

You can change the JMX configuration as follows:

- If you do not want to use the command line interface, and you want to have EDQ JMX Beans appear in the Tomcat application server JMX tree (not the internal JMX server), change the management.port property to 0:

```
management.port=0
```

When `management.port` is set to zero, the RMI registry does not listen on any port. This means that the internal JMX Server will not be used *and* that the RMI API will also not be available. The command line tools will therefore not work if `management.port` is set to 0.

- If you are using Oracle WebLogic Server, and you want to use the command line interface as well as have EDQ JMX Beans appear in the WebLogic Server JMX tree, add the following property to the `director.properties` file in the configuration directory. Retain the setting of 8090 for `management.port` so that the RMI API can be used by the command line tools.

```
management.jndiname=java:comp/env/jmx/runtime
```

Understanding JMX Bean Naming

The naming scheme used for the EDQ JMX Beans is designed to work well with Jconsole. However, other JMX Clients may require a modified naming scheme.

The names used for the EDQ JMX Beans can be customized by writing and placing an appropriate JavaScript or Groovy file in the configuration directory and setting the `management.namemaker.scriptfile` property in the `director.properties` to indicate its existence

Reviewing the Example

This example demonstrates how to modify the default EDQ JMX Bean naming scheme to add a type attribute to the end of the name. The type attribute will be based on the Java Bean class.

1. Create a file named `jmxnames.js` in the configuration directory and add the following JavaScript to it:

```
/**
 * Adds a type attribute to the name of a JMX Beans.
 *
 * @param beanclass The bean class name
 * @param domain The domain name
 * @param names The name strings
 *
 * @return The name string
 */
function objectNameFor(beanclass, domain, names)
{
var type = beanclass == null ? "*" :
beanclass.substring(beanclass.lastIndexOf('.') + 1);
var out;
/*
 * The names array always has 2 elements.
 */
out = domain + ":" + "component=" + escape(names[0]) + ",name=" +
escape(names[1]);
for (var i = 2; i < names.length; i++)
{
var index = i-1
out += "," + "name" + index + "=" + escape(names[i]);
}
return out + ",type=" + type;
}
```

2. Add the following line to the `director.properties` file:

```
management.namemaker.scriptfile = jmxnames.js
```

3. Restart the EDQ application server.

The JMX Beans will now include a type qualifier at the end of their names.

Monitoring Real-Time Processes

EDQ is provided with a built-in JMX server that can be used to monitor many aspects of its operation. Many of the objects and resources that make up the EDQ application provide MBeans to the JMX server, including the real-time Web services.

Monitoring the Real-Time Web Service MBeans

Each real-time Web service registers an MBean for its reader and one for its writer in the JMX tree.

Readers are registered at:

```
Runtime/Data/Buckets/Realtime/Projects/Project Name/readers/Web service name
```

Writers are registered at:

```
Runtime/Data/Buckets/Realtime/Projects/Project Name/writers/Web service name
```

In each case, the path to the MBean includes the name of the Web service that owns it and the project that contains the web service.

Global Web services (those deployed in a `.jar` file in the `oedq_local_home/webservices` directory) have a different path name. Simply replace `Projects/Project Name` in the path above with `Global`.

The port for the internal JMX server is controlled by the `management.port` property, defined in the `director.properties` file.

Monitoring the Real-Time MBeans

A general JMX console, such as JConsole, can be used to interact with MBeans. Each MBean exposes:

- Attributes, whose values can be read.

- Operations that can be invoked to perform some action with the MBean.

- An interface that allows clients to subscribe to notifications of events that occur on the MBean.

The EDQ real-time web service MBeans uses the following attributes:

Attributes	Description
closetime	The time at which the bucket was last closed.
concurrent	The current number of synchronous requests.
maxConcurrent	The maximum number of concurrent synchronous requests since the bucket was opened.
maxConcurrentMax	The maximum number of concurrent synchronous requests since startup.
messages	The number of messages processed since the bucket was opened.
open	Indicates whether the bucket is open or closed.
openCount	The number of times the bucket has been opened since startup.

Attributes	Description
opentime	The time when the bucket was last opened.
processtime	The time when the last message was processed.
records	The number of records processed since the bucket was opened.
threads	The number of threads that used the bucket when it was last opened.
totalMessages	The number of messages processed since startup.
totalRecords	The number of records processed since startup.

The EDQ real-time web service MBeans exposes the following operation:

Attribute	Description
closedown	Shutdown the reader or writer using this bucket.

6
Using Triggers

This chapter describes how to use the trigger functionality in Enterprise Data Quality. This document describes where triggers are installed, how to call them, and how you can use them.

This chapter contains the following topics:

Overview of the Triggers Functionality

Triggers in Enterprise Data Quality are scripts (JavaScript or Groovy) that can be called at various *trigger points* in the EDQ system. There are two types of triggers: predefined triggers and custom triggers.

About Predefined Triggers

Predefined triggers are included with the EDQ installation. They are visible in the Director user interface and can be used in a job configuration to start the job, shut down web services, send email notifications, and run another job from within a job. Director users can set these triggers to run at the following trigger points: the start of a job, the end of a job, or both. You can learn more about predefined triggers in the Director online help system.

About Custom Triggers

Custom triggers can be written by someone skilled in Javascript or Groovy to extend the functionality of EDQ to achieve specific workflow objectives. You can use custom triggers to perform tasks such as:

- sending an email message
- sending a JMS message

- calling a web service

- writing a file

- sending a text message

You can run custom triggers at any of the following predefined trigger points:

- Before running a job phase

- After running a job phase

- On making a match decision

- On making a transition in Case Management

- When a job completes

Each of these trigger point has a unique path and a set of defined arguments that are passed to the trigger through a special API. For more information, see Understanding EDQ Trigger Points.

Custom triggers are described in the rest of this document.

Required Skills to Use Triggers

Knowledge of Javascript or Groovy is required to create and deploy custom triggers in EDQ.

Storing Triggers

Custom triggers must be stored in the `triggers` subdirectory of the EDQ `config` (configuration) directory. New or updated triggers are loaded automatically without requiring a system restart.

Configuring Triggers Using the Script Trigger API

You can use the functions of the script API to create your triggers. These functions are defined in the trigger code. Although the examples in this document are JavaScript, the same API is available in Groovy.

The following are descriptions of each function in this API.

getPath()
Returns a string that defines the path that the trigger will handle. Each trigger point has a unique path. Any trigger that matches a given path is executed when the trigger point is reached. For more information about trigger points, see Understanding EDQ Trigger Points.
This function is a regular expression. For example, the path `/log/com\.datanomic\..*` would match any logging path where the logger name contains the string `datanomic` (in other words, any logger defined in EDQ, the word "datanomic" being another name for EDQ).

run(*path*, *id*, *env*, *arg1*, *arg2* ...)
Executes the trigger. For more information about what is returned by the trigger API for each of these variables, see Understanding EDQ Trigger Points.

path

The path of the trigger, for example `/runtime/engine/interval/end`.

id

The trigger ID. The ID is set when the trigger is configured in the Director user interface. The ID is null for simple triggers.

env

The trigger environment in the form of one or more key/value pairs, for example `env.project = project name`. The `env` input is specific to the trigger point. These values are exposed as properties of the `env` object in the script. Most trigger points will pass in the associated EDQ project ID and project name.

arg

Extra arguments that are specific to the trigger point. For example, the `Interval end` trigger point returns the following: Task context object, process options, interval number (>= 1), execution statistics.

filter(*path*, *env*)

(Optional function) Filters out the trigger before it can be executed. Use this filter to avoid the overhead of executing a trigger that will not be needed. Return `true` to enable the trigger or `false` to disable it.

path

The path of the trigger.

env

The trigger environment in the form of one or more key/value pairs. The `env` input is specific to the trigger point. These values are exposed as properties of the `env` object in the script. Most trigger points will pass in the associated EDQ project ID and project name. In the following example, the trigger is enabled only if the associated project is named "My project."

```
function filter(path, env) {
  return env.project == 'My project';
}
```

getLevel()

(Optional function) Returns the maximum level the trigger will accept. For example, the following statement allows the trigger to accept all levels, regardless of other settings in the trigger system. For more information about setting levels, see Setting Trigger Levels.

```
function getLevel() {
  return Level.SEVERE;
}
```

getTriggerNames(*path*, *env*)

(Optional function) Returns an array of `TriggerName` objects for display in the Director user interface. For more information, see Exposing Triggers in a Job Configuration. Getting trigger names and exposing them in the Director interface is only possible with the job configuration screen.

Extending the Configuration of Triggers Using Properties Files

You can specify additional configuration for script triggers in properties files. Access to these properties is by means of a predefined object named `config`, which is available in all triggers. The base directory in EDQ for these properties files is the subdirectory `config` within the `triggers` directory. The following are useful methods for the `config` object.

config.getTriggerConfigFiles(*base*, *pattern*)
Returns an array of file objects whose names match a search pattern within a specified directory in the `triggers/config` directory.

> **base**
> The name of a directory within the `triggers/config` directory.

> **pattern**
> A regular expression (regex) that defines the search pattern to match.

config.loadProps(*file*)
Loads a specified Java properties file and return it as a JavaScript object.

> **file**
> The name of the Java properties file.

Understanding EDQ Trigger Points

This section describes the trigger points within EDQ at which you can call custom triggers.

Log Message
Called whenever a log message is generated in the system.

Component	Description
Path	/log/*loggername*
Env	null
Arguments	java.util.logging.LogRecord

Syslog Message
Called whenever a high-level `syslog` log message is generated. The *source* argument is a Java object that contains details of the event source. It can be converted to string for display.

Component	Description
Path	/syslog
Env	env.event = *event_name* env.source = *event_source_as_string*
Arguments	*event_name*, *source*, *message*

Process start

Called when a process starts. The arguments are Java objects that contain information on the process configuration.

Component	Description
Path	/runtime/engine/task/start
Env	env.project = *project_name*
	env.projectID = *project_ID*
	env.missionname = *job_name*
	env.processname = *process_name*
Arguments	*Task_context_object*,
	process_options

> **Note:**
>
> When specifying the path for starting a task, the trigger script must include addLibrary('runtime') to avoid the trigger script from throwing an error.

Process end

Called when a process stops. The arguments are Java objects that contain information on the process configuration.

Component	Description
Path	/runtime/engine/task/end
Env	env.project = *project_name*
	env.projectID = *project_ID*
	env.missionname = *job_name*
	env.processname = *process_name*
Arguments	*Task_context_object*,
	process_options

> **Note:**
>
> When specifying the path for ending a task, the trigger script must include addLibrary('runtime') to avoid the trigger script from throwing an error.

Interval end

Called at the end of a normal process or at the end of each interval of a process that is run in interval mode. Returns statistics on the number of records executed, etc.

Component	Description
Path	/runtime/engine/interval/end
Env	env.project = *project_name*
	env.projectID = *project_ID*
	env.missionname = *job_name*
	env.processname = *process_name*

Component	Description
Arguments	*Task_context_object*, *process_options*, *interval_number (>= 1)*, *execution_statistics*

Before job phase
Called in a job configuration for 'pre phase' execution.

Component	Description
Path	/missions/phase/pre
Env	env.project = *project_name* env.projectID = *project_ID* env.missionname = *job_name* env.processname = *process_name*
Arguments	None

After job phase
Called in a job configuration for 'post phase' execution.

Component	Description
Path	/missions/phase/post
Env	env.project = *project_name* env.projectID = *project_ID* env.missionname = *job_name* env.processname = *process_name*
Arguments	None

On match decision
Called when EDQ must make a decision about a potential match. This is known as a *relationship decision* trigger. Relationship triggers can include methods that return the relationship and decision data needed to perform matching. This trigger point is specific to Match Review.

Component	Description
Path	/matchreview/relationship/ decision/
Env	env.project = *project_name*
Arguments	A list of TriggerInfo methods. Each contains data for one relationship. See Understanding TriggerInfo Methods for descriptions of these methods.

When a case is created
Called when a case is created where the case belongs to the respective case source.

Component	Description
Path	/casemanagement/create/<case source name>

Component	Description
Env	env.sourceName = case source name env.caseType = the type of the case or alert env.currentState = the current state of the created case
Arguments	com.datanomic.director.casemanagement.beans.CaseBean

When a case or alert has transitioned
Called after a case or alert has transitioned into the next logical state corresponding to the respective workflow.

Component	Description
Path	/casemanagement/transition/ <workflow name>/<transition>
Env	env.sourceName = case source name env.caseType = the type of the case, 'case' or 'alert' env.currentState = the current state of the created case
Arguments	com.datanomic.director.casemanagement.beans.CaseBean,java.util.List<com.datanomic.director.casemanagement.beans.CaseHistoryBean>,comment,restrictingPermission Where comment is entered by the user and restrictingPermission is the permission required to access the comment.

When a case or alert has been updated
Called when a case or alert is updated by a user. This includes assignment, state change, priority change, or performing any other edit.

Component	Description
Path	/casemanagement/update/<case source name>
Env	env.sourceName = case source name env.caseType = the type of the case, 'case' or 'alert' env.currentState = the current state of the created case

Component	Description
Arguments	com.datanomic.director.casemanagement.beans.CaseBean,java.util.List<com.datanomic.director.casemanagement.beans.CaseHistoryBean>,comment,restrictingPermission Where comment is entered by the user and restrictingPermission is the permission required to access the comment

When a comment is added for a case or alert

Called when a comment is added for a case or alert.

Component	Description
Path	/casemanagement/commented/<case source name>
env.sourceName = case source name Env	env.caseType = the type of the case, 'case' or 'alert' env.currentState = the current state of the created case
Arguments	com.datanomic.director.casemanagement.beans.CaseBean,java.util.List<com.datanomic.director.casemanagement.beans.CaseHistoryBean>,comment,restrictingPermission Where comment is the user entered comment and restrictingPermission is the permission required to access the comment.

After a system update occurs

Called after a case or alert is updated as part of an escalation or bulk update.

Component	Description
Path	/casemanagement/systemupdate/<case source name>
Env	env.sourceName = case source name env.caseType = the type of the case, 'case' or 'alert' env.currentState = the current state of the created case

Component	Description
Arguments	`com.datanomic.director.casemanagement.beans.CaseBean,java.util.List<com.datanomic.director.casemanagement.beans.CaseHistoryBean>,comment,restrictingPermission` Where `comment` is the user entered comment and `restrictingPermission` is the permission required to access the comment.

Understanding TriggerInfo Methods

This section explains each of the methods that are associated with the `TriggerInfo` trigger point. These methods are specific to the `TriggerInfo` trigger point for use in Match Review.

Table 6-1   Methods Associated with the TriggerInfo Trigger Point

Method	Data Returned	Description
`getPreviousMatchStatus()`	String	Returns the match status prior to the decision.
`getPreviousRealtionshipReviewStatus()`	String	Returns the relationship review status prior to the decision.
`getRelationshipId()`	Integer	Returns the relationship ID.
`getRecordId()`	Integer	Returns the ID of the first record.
`getInputId()`	Integer	Returns the ID of the first input.
`getRelatedRecordId()`	Integer	Returns the ID of the second record.
`getRelatedInputId()`	Integer	Returns the ID of the second input.
`getReviewStatus()`	String	Returns the review status of the new relationship.
`getMatchStatus()`	String	Returns the new match status.
`getRuleName()`	String	Returns the name of the rule that generated the relationship.
`getCommentUser()`	String	Returns the user name of the person that made the comment.
`getReviewComment()`	String	Returns any comment that was made.
`getCommentDate()`	Date	Returns the date and time that the comment was made (if comment is present).
`getReviewedUser()`	String	Returns the name of the user who performed the review.
`getReviewDate()`	Date	Returns the date and time that the review was performed.

Table 6-1   (Cont.) Methods Associated with the TriggerInfo Trigger Point

Method	Data Returned	Description
SourceAttribute `getRecordSourceAttributes()`	List	Returns all the source attributes (columns) that make up the first record.
SourceAttribute `getRelatedRecordSourceAttributes()`	List	Returns all the source attributes (columns) that make up the second record.
`getRecordAttributeValue(`*SourceAttribute* `sa)`	Value	Returns the value of the given source attribute (column) of the first record.
`getRelatedRecordAttributeValue(`*SourceAttribute* `sa)`	Value	Returns the value of the given source attribute (column) of the second record.

Setting Trigger Levels

Every trigger point has an associated level, which is a `java.util.logging.Level` value. By default trigger calls with a level lower than `INFO` are ignored.

One way to modify the level is to create a file named `levels.properties` in the `triggers` subdirectory of the `config` directory. This file can contain both a default level and one or more override levels for individual paths. Example 6-1 sets the default level to `FINE` and sets the level for the path `/runtime/engine/.*` to `FINER`. You can define your own prefix for the pattern and level properties.

Another way to modify the level is to define a `getLevel` function in the trigger. See Configuring Triggers Using the Script Trigger API for a description.

Example 6-1   Setting Trigger Levels

```
default = fine
runtime.pattern = /runtime/engine/.*
runtime.level   = finer
```

Using JMS in Triggers

To enable Java Message Service (JMS) within a trigger file, follow these steps.

1. Load the internal JavaScript JMS library.

   ```
   addLibrary("jms");
   ```

2. Load properties that define the JMS configuration. These properties are augmented with the JMS settings from the standard `realtime.properties` file that is shipped in the EDQ configuration directory. The default version of this file defines properties for the open-source ActiveMQ message broker that is bundled with EDQ. At minimum, the trigger should supply a value for the `destination` property, which names the JMS topic or queue to use.

3. Create a JMS object.

   ```
   var jms = JMS.open(props);
   ```

4. Send a text message.

```
jms.send(str)
```

5. Send a JMS map message built from a script object.

```
jms.sendMap(jsobj)
```

6. Create a text message. Properties and header values can be set on the message before transmission.

```
var msg = jms.createTextMessage(str)
```

7. Create a map message. Properties and header values can be set on the message before transmission.

```
var msg = jms.createMapMessage(jsobj)
```

8. Send a message that was created by one of the two preceding methods.

```
jms.sendMessage(msg)
```

Exposing Triggers in a Job Configuration

Triggers are selected for use in a job when configuring a job phase in Director. They can be set to run before or after a job phase. To make triggers available for selection on the configuration screen, each trigger must be able to return a list of names. This allows one trigger to perform multiple tasks as needed.

A trigger name has the following components:

- an internal ID that is passed to the trigger **run** function. See Configuring Triggers Using the Script Trigger API for a description of this function.
- a visible label
- a group name

Trigger names with the same group are shown as a single node in the job configuration screen.

To create a new trigger name:

```
var n1 = new TriggerName(id, label)
n1.group = "My group";
```

To return trigger names from a trigger:

To return trigger names, use the getTriggerNames function as shown in this example.

```
function getTriggerNames(path, env) {
  var n1 = new TriggerName(id1, label1);
  var n2 = new TriggerName(id2, label2);
  ...
  n1.group = "My group";
  n2.group = "My group";
  ...
  return [n1, n2 ...]
}
```

See Configuring Triggers Using the Script Trigger API for more information about getTriggerNames.

Trigger Examples

The following are examples of how you can use custom triggers.

> **✐ Note:**
>
> The examples in this document are JavaScript, but the same API is available in Groovy.

Example 1 Use a Trigger to Send Log Messages Via JMS
In this example, the logging library imports a logging object that can be used to format and output the message. The JMS properties file is loaded from `triggers/config/jms/jms.properties` in the EDQ configuration directory.

```
// Test trigger for task running with JMS
addLibrary("logging");
addLibrary("jms");

function getPath() {
  return "/log/com\.datanomic\..*";
}
function run(path, id, env, logrecord) {

  var pfiles = config.getTriggerConfigFiles("jms",
                             "jms\\.properties");

  if (pfiles.length > 0) {
    var props = config.loadProps(pfiles[0]);

    var jms = JMS.open(props);
    var msg = logging.format(logrecord);
    var len = msg.length;

// Remove trailing newlines

    while (len > 0) {
      var c = msg.charAt(len - 1);

      if (c != '\n' && c != '\r') {
        break;
      }

      len--;
    }

    jms.send(msg.substring(0, len));
    jms.close();
  }
}
```

Example 2 Use a Trigger to Send Syslog Messages Via JMS
In this example, the special `id` directive on the first line (`#! id : syslog`) defines the internal ID of the trigger. If there is more than one trigger definition with the same ID, the later one replaces the former one. In a standard EDQ install, there is a predefined

syslog trigger that logs messages through the standard logging API. Adding the id directive in this example causes the JMS syslog trigger to replace the predefined trigger.

```
#! id : syslog

// Test trigger for task running with JMS

addLibrary("logging");
addLibrary("jms");

function getPath() {
  return "/syslog";
}

function getLevel() {
  return Level.SEVERE;
}

function run(path, id, env, level, event, source, message) {

  var pfiles = config.getTriggerConfigFiles("jms",
                "jms\\.properties");
  var props  = null;

  if (pfiles.length == 0) {
    logger.log(Level.WARNING, "syslogger called but no properties");
  } else {

    props = config.loadProps(pfiles[0]);

    var jms     = JMS.open(props);
    var xml     = <syslog level={level}><source>{source}</source><message>{message}</
message></syslog>

    logger.log(Level.INFO, "xml = {0}", xml.toXMLString());
    jms.send(xml.toXMLString());
    jms.close();
  }
}
```

Example 3 Use a Trigger for Mission Phase Notification
In this example, a couple of trigger names are defined and are exposed to the job configuration screen. The trigger writes a log message in this example, but it could also be configured to send JMS notifications.

```
// Test trigger for misssion phase notification

addLibrary("logging");

function getPath() {
  return "/missions/phase/.*";
}

function run(path, id, env) {
 logger.log(Level.INFO, "phase called with path {0} and id {1}", path, id);
}

function getTriggerNames(path, env) {
```

```
            var n1 = new TriggerName("logme", "logme2");
            n1.group = "logmegroup";

            var n2 = new TriggerName("n2", "n2");
            n2.group = "logmegroup";
            return [n1, n2];
        }
```

7

Accessing EDQ Files Remotely

This chapter describes how to access certain directories in the EDQ directory.

Using FTP and SFTP Server to Access EDQ Files

EDQ is supplied with internal File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP) servers. These servers enable remote access to the configuration file area and landing area files.

The FTP server can be accessed with a third-party FTP client using any valid EDQ username and password, connecting to the port specified by the `ftpserver.port` in the `director.properties` file.

The SFTP server is controlled by the `sshd.port` property in `director.properties`. The default value is 2222.

The following directories are available via the FTP and SFTP servers:

Directory	Description
`config`	This corresponds to the EDQ base configuration directory (`edqhome`).
`config1`	This corresponds to the EDQ local configuration directory (`edq.local.home`).
`landingarea`	This corresponds to the `landingarea` directory in the EDQ installation.
`projectlandingarea`	This corresponds to the project specific landing areas in the EDQ installation.
`commands`	This corresponds to the `commandarea` directory in the EDQ local configuration directory (`oedq_local_home`).

8
Third-Party License Attributions

This chapter describes the legal attributions for third party software and components included in the EDQ product.

Third-Party License Attributions

Apache Active MQ

Apache ActiveMQ Copyright 2006-2008 Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Jabber xsds in activemq-xmpp

This software was generated using XSDs from the Jabber Software Foundation http://www.xmpp.org/schemas/

Copyright (c) 1999-2006 The Jabber Software Foundation

http://www.xmpp.org/about/copyright.shtml

These XSD files are licensed under the Creative Commons License 2.5

http://creativecommons.org/licenses/by/2.5/

==

Jetty Web Container

Copyright 1995-2006 Mort Bay Consulting Pty Ltd

==

This product includes some software developed at The Apache Software Foundation (http://www.apache.org/).

The javax.servlet package used by Jetty is copyright Sun Microsystems, Inc and Apache Software Foundation. It is distributed under the Common Development and Distribution License.

You can obtain a copy of the license at

https://glassfish.dev.java.net/public/CDDLv1.0.html.

The UnixCrypt.java code ~Implements the one way cryptography used by Unix systems for simple password protection. Copyright 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville.

The default JSP implementation is provided by the Glassfish JSP engine from project Glassfish http://glassfish.dev.java.net. Copyright 2005 Sun Microsystems, Inc. and portions Copyright Apache Software Foundation.

Some portions of the code are Copyright:

2006 Tim Vernum

1999 Jason Gilbert.

The jboss integration module contains some LGPL code.

Apache Batik

Copyright 1999-2007 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

This software contains code from the World Wide Web Consortium (W3C) for the Document Object Model API (DOM API) and SVG Document Type Definition (DTD).

This software contains code from the International Organisation for Standardization for the definition of character entities used in the software's documentation.

This product includes images from the Tango Desktop Project (http://tango.freedesktop.org/).

This product includes images from the Pasodoble Icon Theme(http://www.jesusda.com/projects/pasodoble).

Apache Commons

Apache Commons BeanUtils

Copyright 2000-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons Codec

Copyright 2002-2009 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://commons.apache.org/proper/commons-codec/).

----

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java contains test data from http://aspell.sourceforge.net.

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org). Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

Apache Commons Collections

Copyright 2001-2008 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons DBCP

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons Digester

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons FileUpload

Copyright 2002-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache HttpComponents Client

Copyright 1999-2011 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

Apache Commons IO

Copyright 2002-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons Lang

Copyright 2001-2011 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons Logging

Copyright 2003-2007 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Commons Pool

Copyright 2001-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache Jakarta Commons Validator

Copyright 2001-2006 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

===

== NOTICE file corresponding to section 4(d) of the Apache License, ==

== Version 2.0, in this case for the Apache Derby distribution. ==

===

Apache Derby

Copyright 2004-2008 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Portions of Derby were originally developed by International Business Machines Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement", informally known as the "Derby CLA".

The following copyright notice(s) were affixed to portions of the code with which this file is now or was at one time distributed and are placed here unaltered.

(C) Copyright 1997,2004 International Business Machines Corporation. All rights reserved.

(C) Copyright IBM Corp. 2003.

The portion of the functionTests under 'nist' was originally developed by the National Institute of Standards and Technology (NIST), an agency of the United States Department of Commerce, and adapted by International Business Machines Corporation in accordance with the NIST Software Acknowledgment and Redistribution document at http://www.itl.nist.gov/div897/ctg/sql_form.htm

Apache FOP

(C) Copyright 1999-2010 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

This software consists of voluntary contributions made by many individuals on behalf of The Apache Software Foundation and was originally created by James Tauber <jtauber@jtauber.com>.

Apache FTP Server

// --

// NOTICE file corresponding to the section 4d of The Apache License,

// Version 2.0, in this case for Apache FtpServer Parent

// --

Apache FtpServer Parent

Copyright 2003-2010 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache Geronimo

Copyright 2003-2010 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Portions of the Web Console were orginally developed by International Business Machines Corporation and are licensed to the Apache Software Foundation under the "Software Grant and Corporate Contribution License Agreement", informally known as the "IBM Console CLA".

Portions of the ASN1 codec implementation in framework/modules/geronimo-crypto/ were developed by the Bouncy Castle project (http://www.bouncycastle.org/).

Copyright (c) 2000-2005 The Legion Of The Bouncy Castle (http://www.bouncycastle.org)

Apache log4j

Copyright 2007 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Apache Lucene

Copyright 2006 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The snowball stemmers in contrib/snowball/src/java/net/sf/snowball were developed by Martin Porter and Richard Boulton. The full snowball package is available from http://snowball.tartarus.org/

The Arabic stemmer (contrib/analyzers) comes with a default stopword list that is BSD-licensed created by Jacques Savoy. The file resides in contrib/analyzers/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt. See http://members.unine.ch/jacques.savoy/clef/index.html.

The Persian analyzer (contrib/analyzers) comes with a default stopword list that is BSD-licensed created by Jacques Savoy. The file resides in contrib/analyzers/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt. See http://members.unine.ch/jacques.savoy/clef/index.html. Includes lib/servlet-api-2.4.jar from Apache Tomcat

The SmartChineseAnalyzer source code (under contrib/analyzers) was provided by Xiaoping Gao (copyright 2009).

ICU4J, (under contrib/collation) is licensed under an MIT styles license (contrib/collation/lib/ICU-LICENSE.txt) and Copyright (c) 1995-2008 International Business Machines Corporation and others

Apache POI

Copyright 2009 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This product contains the DOM4J library (http://www.dom4j.org). Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved.

This product contains parts that were originally based on software from BEA. Copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

This product contains W3C XML Schema documents. Copyright 2001-2003 (c) World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University)

This product contains the Piccolo XML Parser for Java (http://piccolo.sourceforge.net/). Copyright 2002 Yuval Oren.

This product contains the chunks_parse_cmds.tbl file from the vsdump program. Copyright (C) 2006-2007 Valek Filippov (frob@df.ru)

Apache SSHD

==

== NOTICE file for use with the Apache License, Version 2.0, ==

== in this case for the SSHD distribution. ==

==

This product contains software developped by JCraft,Inc. and subject to the following license:

Copyright (c) 2002,2003,2004,2005,2006,2007,2008 Atsuhiko Yamanaka, JCraft,Inc. All rights reserved.

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.

The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED 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 JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE 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.

Copyright (c) 2000 - 2006 The Legion Of The Bouncy Castle (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache Struts

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Apache v2.0

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of

goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Apache Tomcat

Copyright 1999-2011 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

The Windows Installer is built with the Nullsoft Scriptable Install System (NSIS), which is open source software. The original software and related information is available at http://nsis.sourceforge.net.

Java compilation software for JSP pages is provided by Eclipse, which is open source software. The original software and related information is available at http://www.eclipse.org.

Apache Xalan

==NO TICE file corresponding to the section 4d of the Apache License, Version 2.0, ==

== in this case for the Apache Xalan distribution. ==

==

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

Portions of this software was originally based on the following:

- software copyright (c) 1999-2002, Lotus Development Corporation., http://www.lotus.com.

- software copyright (c) 2001-2002, Sun Microsystems., http://www.sun.com.

- software copyright (c) 2003, IBM Corporation., http://www.ibm.com.

- voluntary contributions made by Ovidiu Predescu (ovidiu@cup.hp.com) on behalf of the Apache Software Foundation and was originally developed at Hewlett Packard Company.

Apache Xerces Java

===

== NOTICE file corresponding to section 4(d) of the Apache License, ==

== Version 2.0, in this case for the Apache Xerces Java distribution. ==

===

Apache Xerces Java

Copyright 1999-2010 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Portions of this software were originally based on the following:

software copyright (c) 1999, IBM Corporation., http://www.ibm.com.

software copyright (c) 1999, Sun Microsystems., http://www.sun.com.

voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

Common Development and Distribution License (CDDL)

The Common Development and Distribution License (CDDL) can be viewed here.

DB Utility Library

Copyright 2005-2010 Richard Evans

This product includes software developed by Richard Evans.

GeoNames

Reference Data in the EDQ Customer Data Services pack (CDS) is partially derived from GeoNames (www.geonames.com).

This data is licensed under the Creative Commons Licence 2.0:

http://creativecommons.org/licenses/by/2.0/

GNU GENERAL PUBLIC LICENSE

The GNU General Public License can be read here.

Groovy Language

==

== NOTICE file corresponding to the section 4 d of ==

== the Apache License, Version 2.0, ==

== in this case for the Groovy Language distribution. ==

==

Groovy Language

Copyright 2003-2009 The respective authors and developers

Developers and Contributors are listed in the project POM file

This product includes software developed by The Groovy community (http://groovy.codehaus.org/).

ICU

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2010 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Janel

Copyright (c) 2003-2007 Timothy D. Kil

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Java Runtime Environment

The Copyright notice for the Java Runtime Environment is reproduced in full below. The Java Runtime Environment also includes a README file and a THIRDPARTYLICENSEREADME file.

Copyright © 2006, 2010, Oracle and/or its affiliates.

All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing. If this is software or related software documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

JavaCC

Copyright (c) 2006, Sun Microsystems, Inc.

All rights reserved.

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 Sun Microsystems, Inc. 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.

JavaHL

This license applies to all portions of Subversion which are not externally-maintained libraries (e.g. apr/, apr-util/, and neon/). Such libraries have their own licenses; we recommend you read them, as their terms may differ from the terms below.

This is version 1 of this license. It is also available online at http://subversion.tigris.org/license-1.html. If newer versions of this license are posted there (the same URL, but with the version number incremented: .../license-2.html, .../license-3.html, and so on), you may use a newer version instead, at your option.

==

Copyright (c) 2000-2006 CollabNet. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. 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.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by CollabNet (http://www.Collab.Net/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The hosted project names must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact info@collab.net.

5. Products derived from this software may not use the "Tigris" name nor may "Tigris" appear in their names without prior written permission of CollabNet.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED 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 COLLABNET OR ITS 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.

==

This software consists of voluntary contributions made by many individuals on behalf of CollabNet.

Jaxen

/*

$Id: LICENSE.txt 1128 2006-02-05 21:49:04Z elharo $

Copyright 2003-2006 The Werken Company. All Rights Reserved.

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 Jaxen Project 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.

*/

JDOM

/*--

$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp $

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (http://www.jdom.org/)." Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/ images/logos.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED 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 JDOM AUTHORS OR THE PROJECT 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.

This software consists of voluntary contributions made by many individuals on behalf of the JDOM Project and was originally created by Jason Hunter <jhunter_AT_jdom_DOT_org> and Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information on the JDOM Project, please see <http://www.jdom.org/>.

*/

JGo

The terms of the Northwoods JGo license require that any company using the JGo software "Include the appropriate Government Restricted Rights legend specified below on all copies of its products incorporating Northwoods technology in accordance with applicable FAR and DFAR regulations." The appropriate Restricted Rights legend is:

U.S. Government Restricted Rights

The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (C)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor / manufacturer is Northwoods Software Corporation, 142 Main St., Nashua, NH 03060.

MySQL JDBC Driver

The following is the contents of the README file supplied with the MySQL JDBC driver. It is also supplied with a copy of the GNU GENERAL PUBLIC LICENSE.

README file

MySQL Connector/J 5.1.15 (formerly MM.MySQL) This is a release of MySQL Connector/J, Oracle's dual-license JDBC Driver for MySQL. For the avoidance of

doubt, this particular copy of the software is released under the version 2 of the GNU General Public License.

MySQL Connector/J is brought to you by the MySQL team at Oracle.

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

CONTENTS

* License

* Documentation Location

* Third-Party Component Notices

LICENSE

The MySQL Connector/J is licensed under the terms of the GPLv2 <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>, like most MySQL Connectors. There are special exceptions to the terms and conditions of the GPLv2 as it is applied to this software, see the FLOSS License Exception <http://www.mysql.com/about/legal/licensing/foss-exception.html>. License information can be found in the COPYING file.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.

DOCUMENTATION LOCATION

The documentation formerly contained in this file has moved into the 'doc' directory, where it is available in HTML, PDF and plaintext forms. You may also find the latest copy of the documentation on the MySQL website at http://dev.mysql.com/doc/refman/5.1/en/connector-j.html

This software is OSI Certified Open Source Software.

OSI Certified is a certification mark of the Open Source Initiative.

****

Third-Party Component Notices

****

This distribution may include materials developed by third parties. For license and attribution notices for these materials, please refer to the documentation that accompanies this distribution. A copy of the license/notices is also reproduced below. GPLv2 Disclaimer For the avoidance of doubt, except that if any license choice other than GPL or LGPL is available it will apply instead, Oracle elects to use only the General Public License version 2 (GPLv2) at this time for any software where a choice of GPL license versions is made available with the language indicating that GPLv2 or any later version may be used, or where a choice of which version of the GPL is applied is otherwise unspecified.

****

%%The following software may be included in this product:

Ant-Contrib

Use of any of this software is governed by the terms of the APACHE SOFTWARE LICENSE.

****

%%The following software may be included in this product: c3p0:JDBC DataSources/ Resource Pools

Use of any of this software is governed by the terms of the GNU LESSER GENERAL PUBLIC LICENSE.

****

%%The following software may be included in this product:

jboss-common-jdbc-wrapper.jar

Use of any of this software is governed by the terms of the GNU LESSER GENERAL PUBLIC LICENSE

****

The following software may be included in this product:

Simple Logging Facade for Java (SLF4J)

Copyright (c) 2004-2008 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

****

PostgreSQL

The PostgreSQL license is available at http://www.postgresql.org/about/licence/.

PostgreSQL JDBC Driver

The PostgreSQL JDBC driver license is available at https://jdbc.postgresql.org/about/
license.html.

Rhino

The majority of Rhino is MPL 1.1 / GPL 2.0 dual licensed:

The MOZILLA PUBLIC LICENSE (http://www.mozilla.org/MPL/MPL-1.1.txt):

The GNU GENERAL PUBLIC LICENSE

==

Additionally, some files (currently the contents of

toolsrc/org/mozilla/javascript/tools/debugger/treetable/) are available

only under the following license:

==

* Copyright 1997, 1998 Sun Microsystems, Inc. All Rights Reserved.

*

* 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 Sun Microsystems 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.

==

Rhino Debugger

The Rhino Debugger license can be accessed at https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino/License.

SLF4J

The SLF4J license can be accessed at http://www.slf4j.org/license.html..

Spring Framework

Spring Framework 3.1

Copyright (c) 2002-2011 SpringSource, a division of VMware, Inc.

This product is licensed to you under the Apache License, Version 2.0 (the "License"). You may not use this product except in compliance with the License. This product may include a number of subcomponents with separate copyright notices and license terms. Your use of the source code for the these subcomponents is subject to the terms and conditions of the subcomponent's license, as noted in the LICENSE file.

Stax License Information

The license information for Stax comprises:

The Binary Code License

The Streaming API for XML (JSR-173) Specification Reference Implementation License

The Third Party License

The Binary Code License

Sun Microsystems, Inc.

Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR, IF THE SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" BUTTON AT THE END OF THIS AGREEMENT.

1. LICENSE TO USE. Sun grants you a non-exclusive and non-transferable license for the internal use only of the accompanying software and documentation and any error corrections provided by Sun (collectively "Software"), by the number of users and the class of computer hardware for which the corresponding fee has been paid.

2. RESTRICTIONS. Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Except as specifically authorized in any Supplemental License Terms, you may not make copies of Software, other than a single copy of Software for archival purposes. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. Licensee acknowledges that Licensed Software is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun Microsystems, Inc. disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or its licensors is granted under this Agreement.

3. LIMITED WARRANTY. Sun warrants to you that for a period of ninety (90) days from the date of purchase, as evidenced by a copy of the receipt, the media on which Software is furnished (if any) will be free of defects in materials and workmanship under normal use. Except for the foregoing, Software is provided "AS IS". Your exclusive remedy and Sun's entire liability under this limited warranty will be at Sun's option to replace Software media or refund the fee paid for Software.

4.DISCLAIMER OF WARRANTY. UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

5.LIMITATION OF LIABILITY. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event will Sun's liability to you, whether in contract, tort (including negligence), or otherwise, exceed the amount paid by you for Software under this Agreement. The foregoing limitations will apply even if the above stated warranty fails of its essential purpose.

6.Termination. This Agreement is effective until terminated. You may terminate this Agreement at any time by destroying all copies of Software. This Agreement will terminate immediately without notice from Sun if you fail to comply with any provision of this Agreement. Upon Termination, you must destroy all copies of Software.

7. Export Regulations. All Software and technical data delivered under this Agreement are subject to US export control laws and may be subject to export or import regulations in other countries. You agree to comply strictly with all such laws and regulations and acknowledge that you have the responsibility to obtain such licenses to export, re-export, or import as may be required after delivery to you.

8.U.S. Government Restricted Rights.If Software is being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), then the Government's rights in Software and accompanying documentation will be only as set forth in this Agreement; this is in accordance with 48 CFR 227.7201

through 227.7202-4 (for Department of Defense (DOD) acquisitions) and with 48 CFR 2.101 and 12.212 (for non-DOD acquisitions).

9. Governing Law. Any action related to this Agreement will be governed by California law and controlling U.S. federal law. No choice of law rules of any jurisdiction will apply.

10.Severability. If any provision of this Agreement is held to be unenforceable, this Agreement will remain in effect with the provision omitted, unless omission would frustrate the intent of the parties, in which case this Agreement will immediately terminate.

11.Integration. This Agreement is the entire agreement between you and Sun relating to its subject matter.It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

JAVA OPTIONAL PACKAGE

SUN JAVA STREAMING XML PARSER, VERSION 1.0

SUPPLEMENTAL LICENSE TERMS

These supplemental license terms ("Supplemental Terms") add to or modify the terms of the Binary Code License Agreement (collectively, the "Agreement"). Capitalized terms not defined in these Supplemental Terms shall have the same meanings ascribed to them in the Agreement. These Supplemental Terms shall supersede any inconsistent or conflicting terms in the Agreement, or in any license contained within the Software.

A. Software Internal Use and Development License Grant. Subject to the terms and conditions of this Agreement, including, but not limited to Section 3 Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce internally and use internally the binary form of the Software, complete and unmodified, for the sole purpose of designing, developing and testing your Java applets and applications ("Programs").

B. License to Distribute Software. In addition to the license granted in Section 1 (Software Internal Use and Development License Grant) of these Supplemental Terms, subject to the terms and conditions of this Agreement, including but not limited to, Section 3 (Java Technology Restrictions) of these Supplemental Terms, Sun grants you a non-exclusive, non-transferable, limited license to reproduce and distribute the Software in binary code form only, provided that you (i) distribute the Software complete and unmodified and only bundled as part of your Programs, (ii) do not distribute additional software intended to replace any component(s) of the Software, (iii) do not remove or alter any proprietary legends or notices contained in the Software, (iv) only distribute the Software subject to a license agreement that protects Sun's interests consistent with the terms contained in this Agreement, and (v) agree to defend and indemnify Sun and its licensors from and against any damages, costs, liabilities, settlement amounts and/or expenses (including attorneys' fees) incurred in connection with any claim, lawsuit or action by any third party that arises or results from the use or distribution of any and all Programs and/or Software.

C. Java Technology Restrictions. You may not create, modify, or change the behavior of, or authorize your licensees to create, modify, or change the behavior of, classes,

interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Sun in any naming convention designation.

D. Trademarks and Logos. You acknowledge and agree as between you and Sun that Sun owns the SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET trademarks and all SUN, SOLARIS, JAVA, JINI, FORTE, and iPLANET-related trademarks, service marks, logos and other brand designations ("Sun Marks"), and you agree to comply with the Sun Trademark and Logo Usage Requirements currently located at http://www.sun.com/policies/trademarks. Any use you make of the Sun Marks inures to Sun's benefit.

E. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of this Agreement. Source code may not be redistributed unless expressly provided for in this Agreement.

F. Third Party Code. Additional copyright notices and license terms applicable to portions of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file. In addition to any terms and conditions of any third party opensource/freeware license identified in the THIRDPARTYLICENSEREADME.txt file, the disclaimer of warranty and limitation of liability provisions in paragraphs 5 and 6 of the Binary Code License Agreement shall apply to all Software in this distribution.

G. Termination for Infringement. Either party may terminate this Agreement immediately should any Software become, or in either party's opinion be likely to become, the subject of a claim of infringement of any intellectual property right.

For inquiries please contact: Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054.

(Form ID#011801)

The Streaming API for XML (JSR-173) Specification Reference Implementation License

Streaming API for XML (JSR-173) Specification

Reference Implementation

License Agreement

READ THE TERMS OF THIS (THE "AGREEMENT") CAREFULLY BEFORE VIEWING OR USING THE SOFTWARE LICENSED HEREUNDER. BY VIEWING OR USING THE SOFTWARE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO ORIGINAL CONTRIBUTOR, DEFINED HEREIN.

1.0 DEFINITIONS.

1.1. "BEA" means BEA Systems, Inc., the licensor of the Original Code.

1.2. "Contributor" means BEA and each entity that creates or contributes to the creation of Modifications.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof and corresponding documentation released with the source code.

1.4. "Executable" means Covered Code in any form other than Source Code.

1.5. "FCS" means first commercial shipment of a product.

1.6. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

(a) Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

(b) Any new file that contains any part of the Original Code or previous Modifications.

1.7. "Original Code" means Source Code of computer software code Reference Implementation.

1.8. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent for which the grantor has the right to grant a license.

1.9. "Reference Implementation" means the prototype or "proof of concept" implementation of the Specification developed and made available for license by or on behalf of BEA.

1.10. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated documentation, interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice.

1.11. "Specification" means the written specification for the Streaming API for XML, Java technology developed pursuant to the Java Community Process.

1.12. "Technology Compatibility Kit" or "TCK" means the documentation, testing tools and test suites associated with the Specification as may be revised by BEA from time to time, that is provided so that an implementer of the Specification may determine if its implementation is compliant with the Specification.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this Agreement or a future version of this Agreement issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2.0 SOURCE CODE LICENSE.

2.1. Copyright Grant. Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Covered Code of such Contributor, if any, and such derivative works, in Source Code and Executable form.

2.2. Patent Grant. Subject to the terms of this Agreement, each Contributor hereby grants You a non-exclusive, worldwide, royalty-free patent license under the Patent Claims to make, use, sell, offer to sell, import and otherwise transfer the Covered Code prepared and provided by such Contributor, if any, in Source Code and

Executable form. This patent license shall apply to the Covered Code if, at the time a Modification is added by the Contributor, such addition of the Modification causes such combination to be covered by the Patent Claims. The patent license shall not apply to any other combinations which include the Modification.

2.3. Conditions to Grants. You understand that although each Contributor grants the licenses to the Covered Code prepared by it, no assurances are provided by any Contributor that the Covered Code does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to You for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, You hereby assume sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow You to distribute Covered Code, it is Your responsibility to acquire that license before distributing such code.

2.4. Contributors' Representation. Each Contributor represents that to its knowledge it has sufficient copyright rights in the Covered Code it provides, if any, to grant the copyright license set forth in this Agreement.

3.0 DISTRIBUTION RESTRICTIONS.

3.1. Application of Agreement.

The Modifications which You create or to which You contribute are governed by the terms of this Agreement, including without limitation Section 2.0. The Source Code version of Covered Code may be distributed only under the terms of this Agreement or a future version of this Agreement released under Section 6.1, and You must include a copy of this Agreement with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this Agreement or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.3.

3.2. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by BEA and including the name of BEA in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.3. Required Notices.

You must duplicate the following notice in each file of the Source Code:

"(c) 2002, 2003 BEA Systems, Inc. All rights Reserved. The contents of this file are subject to the BEA Streaming API for XML Specification Reference Implementation License Agreement (the "Agreement"); you may not use this file except in compliance with the Agreement. A copy of the Agreement is available at http://www.bea.com/"

If You created one or more Modification(s) You may add your name as a Contributor to the copyright portion of the notice above. You must also duplicate this Agreement in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of BEA or any other Contributor. You must make it absolutely clear than any such

warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify BEA and every other Contributor for any liability incurred by BEA or such other Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.4. Distribution of Executable Versions.

You may choose to distribute Covered Code in Executable form under its own license agreement, provided that:

(a) You comply with the terms and conditions of this Agreement; and

(b) Your license agreement: (i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose; (ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits; (iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and (iv) states that Source Code for the Covered Code is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

(c) You do not use any marks, brands or logos associated with the JCP Specification, or otherwise promote or market any Covered Code, as being compatible, compliant, conformant or otherwise consistent with the Specification unless such product passes, in accordance with the documentation (including the TCK Users Guide, if any), the most current TCK applicable to the latest version of the Specification and available from BEA one hundred twenty (120) days before FCS of such version of the product; provided, however, that if You elect to use a version of the TCK also provided by BEA that is newer than that which is required under this Section 2.1(b)(v), then You agree to pass such TCK.

3.5. Distribution of Source Code Versions.

When You make Covered Code available in Source Code form:

(a) it must be made available under this Agreement; and

(b) a copy of this Agreement must be included with each copy of the Covered Code.

You may not remove or alter any copyright notices contained within the Covered Code. Each Contributor must identify itself as the originator of its contribution to the Covered Code, if any, in a manner that reasonably allows subsequent licensees to identify the originator of each portion of the Covered Code.

4.0 DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT BEA OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

5.0 TERMINATION.

5.1. This Agreement and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this Agreement. Provisions which, by their nature, must remain in effect beyond the termination of this Agreement shall survive.

5.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against BEA or a Contributor (BEA or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Covered Code directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.0of this Agreement shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Covered Code against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Section 2.0 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Covered Code, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.0 are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

5.3. If You assert a patent infringement claim against Participant alleging that such Participant's Covered Code directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.0 shall be taken into account in determining the amount or value of any payment or license.

5.4. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

6.0 LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOUBEA, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

7.0 U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

8.0 MISCELLANEOUS.

This Agreement represents the complete agreement concerning subject matter hereof. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this Agreement shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this Agreement.

9.0 RESPONSIBILITY FOR CLAIMS.

As between BEA and the other Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this Agreement and You agree to work with BEA and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

Sequence Library License for SVN

Sequence Library License

This license applies to all portions of the Sequence library, which are not externally-maintained libraries (e.g. junit or jsch).

===

Copyright (c) 2000-2008 SyntEvo GmbH, Ainring, GERMANY.

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

1. Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

2. 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.

3. The end-user documentation included with the redistribution, if

any, must include the following acknowledgment: "This product includes

software developed by SyntEvo GmbH, Ainring, GERMANY."

Alternately, this acknowledgment may appear in the software itself, if

and wherever such third-party acknowledgments normally appear.

4. The hosted project names must not be used to endorse or promote

products derived from this software without prior written

permission. For written permission, please contact info@syntevo.com.

5. Neither the name of SyntEvo GmbH 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 ``AS IS'' AND ANY EXPRESSED 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 SyntEvo GmbH OR HIS 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.

==

The Third Party License

%% The following software may be included in this product: Xerces. Use of any of this software is governed by the terms of the Apache Software License.

* This software consists of voluntary contributions made by many

* individuals on behalf of the Apache Software Foundation. For more

* information on the Apache Software Foundation, please see

* <http://www.apache.org/>.

*

* Portions of this software are based upon public domain software

* originally written at the National Center for Supercomputing Applications,

* University of Illinois, Urbana-Champaign.

*/

Apache Software License

DO NOT TRANSLATE OR LOCALIZE.

* The Apache Software License, Version 1.1

*

* Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

* notice, this list of conditions and the following disclaimer.

*

* 2. 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.

*

* 3. The end-user documentation included with the redistribution, if

* any, must include the following acknowledgment:

* "This product includes software developed by the

* Ant-Contrib project (http://sourceforge.net/projects/ant-contrib)."

* Alternately, this acknowledgment may appear in the software itself,

* if and wherever such third-party acknowledgements normally appear.

*

* 4. The name Ant-Contrib must not be used to endorse or promote products

* derived from this software without prior written permission. For

* written permission, please contact

* ant-contrib-developers@lists.sourceforge.net.

*

* 5. Products derived from this software may not be called "Ant-Contrib"

* nor may "Ant-Contrib" appear in their names without prior written

* permission of the Ant-Contrib project.

*

* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED 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 ANT-CONTRIB

* PROJECT OR ITSCONTRIBUTORS 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.

* ==

GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you want), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software. Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We want to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you". Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify,

sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later

version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you want to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail. If the program is interactive, make it output a short notice like this when it starts in an interactive mode: Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.