

# Chapter 18: Form and Inquiry Configuration Files

The configuration files for the inquiry and forms features are provided in **XML** format. In this chapter we illustrate the structure of these configuration files.

We recommend that you read and understand the material in Chapter 16: *Introduction to XML and DTD Files* before delving into the material presented here. Also see Chapter 11: *Configuring and Launching CRL* for a task-oriented approach to configuring the application; it refers you to task-appropriate sections in the current and other chapters.

Some of the XML configuration files listed in this chapter are provided with **CRL** as samples. They may be used as is, edited, copied, or ignored!

## 18.1 Form Definition Files

---

The Logbook form definition files govern the form entry types. These files must be located in a directory defined by the `Logbook.file_location.forms_directory` parameter in the properties file (see Chapter 15: *CRL's Java Properties*). The `form.dtd` file must also be present in that directory. The **CRL** administrator can create new forms and add them to the same directory, as long as the XML code contains only valid elements. See section 11.5 *Creating Configuration Files for Forms*.

### 18.1.1 XML Elements Allowed in Forms

Unlike the other configuration files in **CRL**, the form definition XML files are not strictly governed by the DTD file, but they do reference it. We have turned off case-sensitivity for form elements and attributes. For consistency and completeness, we list the valid elements in the form of a DTD file. See section 16.3 *Element Types and Attributes in the DTD File* to understand the relationship between DTD and XML files.

The following listing is not guaranteed accurate for **CRL v1\_7\_07** (we have not generated a new DTD listing for this version); this is approximate only:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!ELEMENT Form ( Line+, REPEATBLOCK*, insertform* ) >
```

```

<!ATTLIST Form name CDATA #IMPLIED >
<!ATTLIST Form reload NMOKEN #IMPLIED >
<!ATTLIST Form Colspan CDATA #IMPLIED >

<!ELEMENT insertform EMPTY >
<!ATTLIST insertform name NMOKEN #REQUIRED >
<!ATTLIST insertform reload NMOKEN #IMPLIED >
<!ATTLIST insertform byReference NMOKEN #IMPLIED >

<!ELEMENT Line ( Text*, Field*, DateAndTime*, CheckBox*, RadioButtonGroup*, Select*,
List*, TABLE*, EXEC* ) >
<!ATTLIST Line align NMOKEN #IMPLIED >

<!ELEMENT Text ( #PCDATA ) >
<!ATTLIST Text Colspan CDATA #IMPLIED >

<!ELEMENT CheckBox EMPTY >
<!ATTLIST CheckBox Name CDATA #REQUIRED >
<!ATTLIST CheckBox checked NMOKEN #REQUIRED >
<!ATTLIST CheckBox Colspan CDATA #IMPLIED >

<!ELEMENT DateAndTime EMPTY >
<!ATTLIST DateAndTime Date NMOKEN #REQUIRED >
<!ATTLIST DateAndTime Time NMOKEN #REQUIRED >
<!ATTLIST DateAndTime Colspan CDATA #IMPLIED >

<!ELEMENT List ( Text*, Item* ) >
<!ATTLIST List Multiple NMOKEN #REQUIRED >
<!ATTLIST List Colspan CDATA #IMPLIED >

<!ELEMENT Item EMPTY >
<!ATTLIST Item Name CDATA #REQUIRED >

<!ELEMENT Field EMPTY >
<!ATTLIST Field Columns NMOKEN #IMPLIED >
<!ATTLIST Field Rows NMOKEN #IMPLIED >
<!ATTLIST Field Colspan CDATA #IMPLIED >

<!ELEMENT Group EMPTY >
<!ATTLIST Group Colspan CDATA #IMPLIED >

<!ELEMENT Option EMPTY >
<!ATTLIST Option Name CDATA #REQUIRED >

<!ELEMENT Select ( Text*, Option* ) >
<!ATTLIST Select Editable NMOKEN #IMPLIED >

<!ELEMENT RadioButtonGroup ( Text*, RadioButton* ) >
<!ATTLIST RadioButtonGroup Colspan CDATA #IMPLIED >

<!ELEMENT RadioButton EMPTY >
<!ATTLIST RadioButton Name NMOKEN #REQUIRED >
<!ATTLIST RadioButton checked NMOKEN #REQUIRED >

<!ELEMENT TABLE ( ColumnLabel* ) >
<!ATTLIST Field width CDATA #IMPLIED >
<!ATTLIST Field height CDATA #IMPLIED >

<!ELEMENT EXEC EMPTY >
<!ATTLIST Program CDATA #REQUIRED >

<!ELEMENT ColumnLabel ( DateAndTime*, CheckBox*, Integer*, Double*, Select*, Field* )
>
<!ATTLIST ColumnLabel name CDATA #REQUIRED >

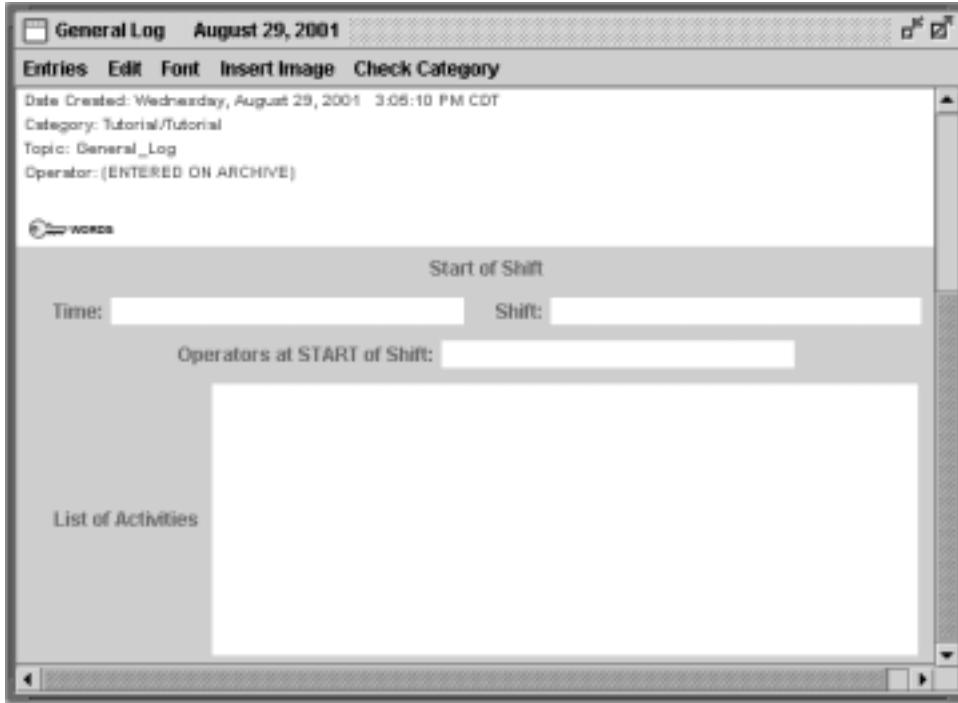
```

```
<!ELEMENT Integer EMPTY >  
<!ELEMENT Double EMPTY >  
<!ELEMENT REPEATBLOCK ( Line* ) >
```

### 18.1.2 Sample “Start of Shift” Form

```
<?xml version="1.0"?>  
<!DOCTYPE Form SYSTEM "form.dtd">  
<Form name="Start of Shift">  
    <Line>  
        <Text>Time:</Text>  
        <Field Rows="1" Columns="20" />  
        <Text>      Shift:</Text>  
        <Field Rows="1" Columns="21" />  
    </Line>  
    <Line>  
        <Text>Operators at START of Shift:</Text>  
        <Field Rows="1" Columns="20" />  
    </Line>  
    <Line>  
        <Text>List of Activities </Text>  
        <Field Rows="10" Columns="40" />  
    </Line>  
</Form>
```

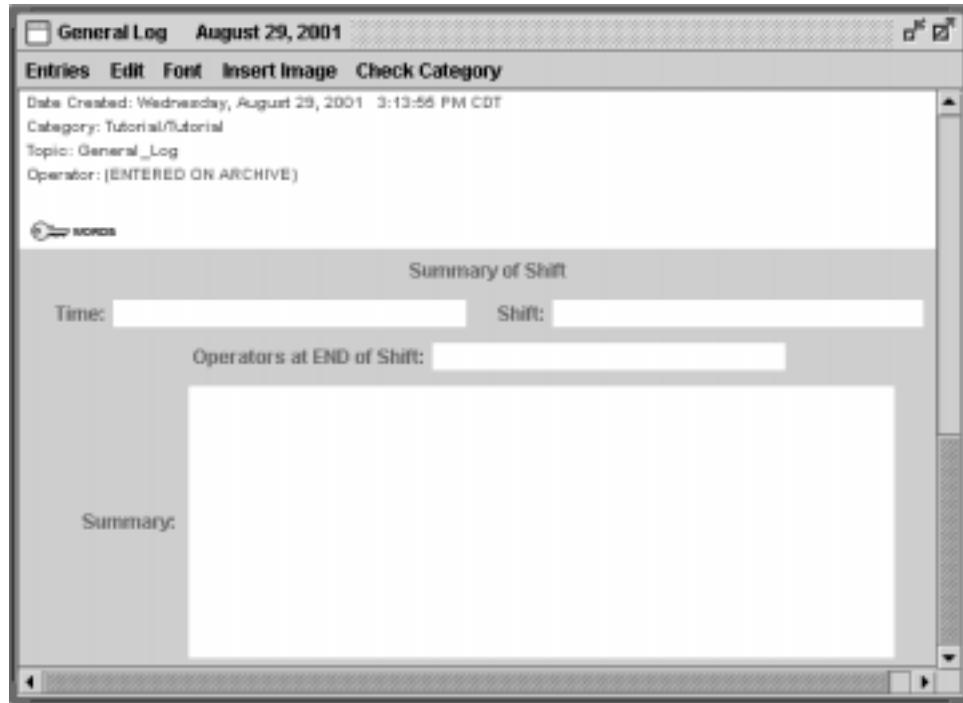
This XML file produces the following **START OF SHIFT** form:



### 18.1.3 Sample “End of Shift” Form

```
<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="Summary of Shift">
    <Line>
        <Text>Time:</Text>
        <Field Rows="1" Columns="20" />
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>
    <Line>
        <Text>Operators at END of Shift:</Text>
        <Field Rows="1" Columns="20" />
    </Line>
    <Line>
        <Text>Summary: </Text>
        <Field Rows="10" Columns="40" />
    </Line>
</Form>
```

This XML file produces the following **SUMMARY OF SHIFT** form:



#### 18.1.4 Sample Form using a Variety of Form Elements

This example shows how to implement list boxes, select boxes (dropdown lists), check boxes, radio buttons, program output and date and time in your forms. This file does not come with the default configuration.

Insert the initial lines, plus Form name:

```
<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="Start of Run">
```

Insert two Date/Time fields; one for date only and one for time only:

```
<Line align="left">
  <Text>Date:</Text>
  <DateAndTime Date="yes" Time="no" />
  <Text>  Time:</Text>
  <DateAndTime Date="no" Time="yes" />
```

On the same line, insert a select box for Shift:

```
<Select>
  <Text> Shift: </Text>
  <Option Name="Day" />
  <Option Name="Evening" />
  <Option Name="Night" />
</Select>
```

```
</Line>
```

On the next line, insert an editable select box:

```
<Line align="left">
    <Select Editable="yes">
        <Text>Select Box:  </Text>
        <Option Name="John Q. Smith" />
        <Option Name="Pierre Auguste Renoir" />
        <Option Name="Claude Monet" />
        <Option Name="Edouard Manet" />
        <Option Name="Edgar Degas" />
    </Select>
```

On the same line, insert two radio buttons, and set one to a default initial value of “on”, the other to “off”; user can select only one of the set:

```
<RadioButtonGroup>
    <Text>      Radio Button:  </Text>
    <RadioButton Name="ON" checked="on" />
    <RadioButton Name="OFF" checked="off" />
</RadioButtonGroup>
</Line>
```

Insert some more check boxes, and set each to a default initial value of “on” or “off”:

```
<Line align="left">
    <Text>Checkbox:  </Text>
    <CheckBox Name="Fast IO" checked="on" />
    <CheckBox Name="Fancy Graphics" checked="on" />
    <CheckBox Name="High Bandwidth" checked="off" />
</Line>
```

Insert a text box:

```
<Line align="left">
    <Text>Text field, 1 row 10 columns:</Text>
    <Field Rows="1" Columns="10" />
</Line>
```

Insert a list box:

```
<Line align="left">
    <List Multiple="yes">
        <Text>List Box (Impressionists):  </Text>
        <Item Name="Van Gogh"/>
        <Item Name="Renoir"/>
        <Item Name="Gauguin"/>
        <Item Name="Monet"/>
    </List>
</Line>
```

Run a program and insert the (text) output into a text area which sizes itself automatically (insert a header line first):

```

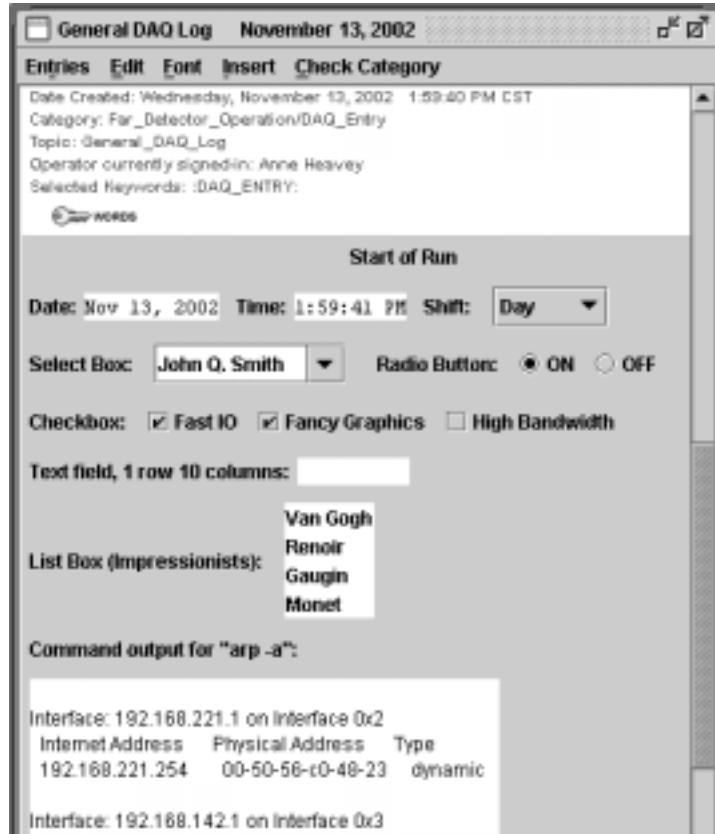
<Line align="left">
    <Text>Command output for "arp -a":</Text>
</Line>
<Line align="left">
    <exec Program="arp -a" />
</Line>

```

Close form:

```
</Form>
```

This XML file produces the following **START OF RUN** form:



### 18.1.5 Sample Table Element in a Form

Note that the table element is contained within a line element. The table height and width default to 650 and 150, respectively.

```

<Line>
    <Table height="50" width="200">
        <ColumnLabel name="Date">
            <DateAndTime Date="yes" Time="no" />
        </ColumnLabel>
        <ColumnLabel name="Check Box">
            <CheckBox checked="on" />

```

```

    </ColumnLabel>
    <ColumnLabel name="Number int">
        <Integer/>
    </ColumnLabel>
    <ColumnLabel name="Number float">
        <Double/>
    </ColumnLabel>
    <ColumnLabel name="Select Box">
        <Select Editable="yes">
            <Option Name="CMSSKIN"/>
            <Option Name="CMSIM"/>
            <Option Name="OOHIT"/>
            <Option Name="OODIGIS"/>
            <Option Name="NTUPLE"/>
        </Select>
    </ColumnLabel>
    <ColumnLabel name="Text">
        <Field/>
    </ColumnLabel>
</Table>
</Line>

```

[Table Headers]					
[Buttons: Add New Row, Delete Selected Row]		[Table Columns: Date, Check Box, Number int, Number float, Select Box, Text]			
1:49:40 PM	<input type="checkbox"/>	0	0	CMSIM	
[Empty Row]					

### 18.1.6 Sample Line for Running a Script or Command

You can run a program within a form using the `EXEC` element with its `Program` attribute. The program can be virtually any type of script (e.g., a Python script), or an OS command. In this example, we execute the command `arp -a`:

```

<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="...">
    ...
    <Line>
        <EXEC Program="arp -a" />
    </Line>
    ...
</Form>

```

The form automatically sizes the field to accommodate the output.

### 18.1.7 Sample Lines with Field Placement Grouping

The element `<GROUP>` is meant to group form fields for neat and regular placement in the HTML file created from the entry as viewed on the Web. It does not affect placement of the fields as seen in a container within **CRL**.

Each XML `<LINE>` element translates into an HTML `<TR>` (table row). Each `<GROUP>` element within a `<LINE>` translates into a `<TD>` (a table cell).

The first line in your form in which you use grouping sets the number and (possibly varying) widths of columns in the table. It's best not to use the `<COLSPAN>` attribute in the first line. Subsequent lines may use grouping with `<COLSPAN>` to place form fields horizontally relative to the first line. For example, the following XML code:

```
<line>
  <group>
    <text>field1 </text>
    <text>field2 </text>
  </group>

  <group>
    <text>field3 </text>
    <text>field4 </text>
  </group>

  <group>
    <text>field5 </text>
    <text>field6 </text>
  </group>
</line>

<line>
  <group colspan="3">
    <text>field7 </text>
    <text>field8 </text>
    <text>field9 </text>
    <text>field10 </text>
    <text>field11 </text>
  </group>
</line>
```

...produces two lines, the first with three sets of two grouped fields, and the second with one set of five grouped fields which should span the same width of three columns as the first line. The HTML should appear on the browser as follows:

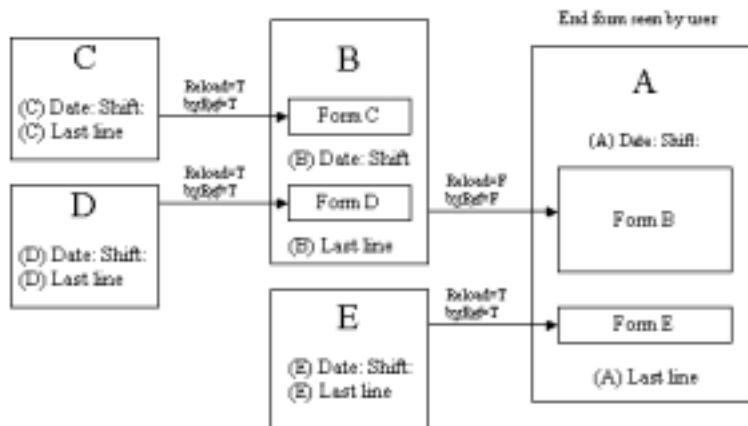
```
field1 field2      field3 field4      field5 field6
field7     field8      field9     field10    field11
```

If the field contents are of varying lengths, the result may be less regular than that shown. You may have to test your form a few times to get it the way you want it to look.

The `<COLSPAN>` attribute can be used with the following elements: CheckBox, DateAndTime, Field, Form, Group, List, RadioButtonGroup, and Text. In all cases, it sets the column span for the contents of the field as viewed in HTML.

### 18.1.8 Sample Form with Embedded Forms

These sample forms provide details for the example discussed in section 11.5.4 *Create Forms with Selected Reloadable Fields*. We recommend that you read that section first. The schematic below illustrates our example:



#### Form A (The End Form)

This form has two native elements, the Date/Time and Shift line and the LAST LINE. Inbetween these two elements, it inserts two forms, b.xml and e.xml (for B and E), the former with reload and byReference false, and the latter with reload and byReference true.

```
<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="AAA">
<Line>
<Text>Date:</Text>
```

```

        <DateAndTime Date="yes" Time="no" />
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>
<insertform name="b.xml" reload="false" byReference="false"/>
<insertform name="e.xml" reload="true" byReference="true"/>
    <Line>
        <Text>LAST LINE OF AAA</Text>
        <Field Rows="1" Columns="20" />
    </Line>
</Form>

```

## Form B (b.xml)

This form has two native elements, the Date/Time and Shift line and the LAST LINE. Above and between these two elements, it inserts two forms, c.xml and d.xml (for C and D), both with reload and byReference true. Form B gets inserted into A with reload and byReference both false.

```

<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="BBB">
<insertform name="c.xml" reload="true" byReference="true"/>
    <Line>
        <Text>Date:</Text>
        <DateAndTime Date="yes" Time="yes" />
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>
<insertform name="d.xml" reload="true" byReference="true"/>
    <Line>
        <Text>LAST LINE OF BBB</Text>
        <Field Rows="1" Columns="20" />
    </Line>
</Form>

```

## Form C (c.xml)

This form has two native elements only, the Date/Time and Shift line and the LAST LINE. It gets inserted into B with reload and byReference both true.

```

<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="CCC">
    <Line>
        <Text>Date:</Text>
        <DateAndTime Date="yes" Time="yes" />
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>

```

```

</Line>
<Line>
    <Text>LAST LINE OF CCC</Text>
    <Field Rows="1" Columns="20" />
</Line>
</Form>

```

## Form D (d.xml)

This form has two native elements only, the Date/Time and Shift line and the LAST LINE. It gets inserted into B with reload and byReference both true.

```

<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="DDD">
    <Line>
        <Text>Date:</Text>
        <DateAndTime Date="yes" Time="yes"/>
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>
    <Line>
        <Text>LAST LINE OF DDD</Text>
        <Field Rows="1" Columns="20" />
    </Line>
</Form>

```

## Form E (e.xml)

This form has two native elements only, the Date/Time and Shift line and the LAST LINE. It gets inserted into A with reload and byReference both true.

```

<?xml version="1.0"?>
<!DOCTYPE Form SYSTEM "form.dtd">
<Form name="EEE">
    <Line>
        <Text>Date:</Text>
        <DateAndTime Date="yes" Time="yes"/>
        <Text>      Shift:</Text>
        <Field Rows="1" Columns="21" />
    </Line>
    <Line>
        <Text>LAST LINE OF EEE</Text>
        <Field Rows="1" Columns="20" />
    </Line>
</Form>

```

## How does this look to the user?

First, user creates a logbook entry of the form A type on June 13 at 4:18 p.m. Initially, it looks like this (ignore the aging message):

General Log - June 13, 2002  
Delete Edit Insert Check Category

THIS ENTRY IS AGING. Consider saving or deleting.

Date: June 13, 2002 Time: 4:18:13 PM  
Name: SAMS Address: 123 Main Street, City: Anytown, State: Any, Zip: 12345  
Phone: (123)456-1234 Email: sams@anywhere.com  
Email: sams@anywhere.com

CCC

Date: June 13, 2002 4:18:13 PM Name: SAMS Address: 123 Main Street, City: Anytown, State: Any, Zip: 12345  
Phone: (123)456-1234 Email: sams@anywhere.com

SDO

Date: June 13, 2002 4:18:13 PM Name: SAMS Address: 123 Main Street, City: Anytown, State: Any, Zip: 12345  
Phone: (123)456-1234 Email: sams@anywhere.com

WWD

Date: June 13, 2002 4:18:13 PM Name: SAMS Address: 123 Main Street, City: Anytown, State: Any, Zip: 12345  
Phone: (123)456-1234 Email: sams@anywhere.com

SFE

Date: June 13, 2002 4:18:13 PM Name: SAMS Address: 123 Main Street, City: Anytown, State: Any, Zip: 12345  
Phone: (123)456-1234 Email: sams@anywhere.com

AAA

Buttons: (1) Exit without saving changes (2) Save changes

Look at the date/time entries. Forms A and B have the current date/time (B was inserted with reload=false). Forms C, D, and E have an earlier time (they were each reloaded upon insertion into target form).

The user now edits the fields (leaving the date/time fields alone), and archives. The form now looks like this:

General Log - June 13, 2002

**Entries**    **Exit**    **Sort**    **Reset**    **Check Category**

Normal Location: D:\work\2002\4\Logs\114\Logs\Other\Relocation\RelocationLog.htm (Line 1, 100)

Date Created: Thursday, June 13, 2002 4:28:41 PM (PT)

Date Entered: Thursday, June 13, 2002 4:27:02 PM (PT)

Processor Number: 100

Category: Relocation

Type: General Log

Operator: User Log

Selected Records: 0

**AAA**

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

000

000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

LAST LINE OF 000: 000000000000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

LAST LINE OF AAA: 000000000000

Status: 110 entries are currently selected

Now, user creates another form A. It looks like this initially:

General Log - June 13, 2002

**Entries**    **Exit**    **Sort**    **Reset**    **Check Category**

Normal Location: D:\work\2002\4\Logs\114\Logs\Other\Relocation\RelocationLog.htm (Line 1, 100)

Date Created: Thursday, June 13, 2002 4:28:41 PM (PT)

Date Entered: Thursday, June 13, 2002 4:27:02 PM (PT)

Processor Number: 100

Category: Relocation

Type: General Log

Operator: User Log

Selected Records: 0

**AAA**

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

000

000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

LAST LINE OF 000: 000000000000

000

Date: Jun 13, 2002 4:44:10 PM    1000: 000000000000

LAST LINE OF 000: 000000000000

LAST LINE OF AAA: 000000000000

Status: 110 entries are currently selected

Notice that A and B again reflect an updated time (well, you can't actually tell for A since the time doesn't show) and that the other A and B fields are cleared. E, C, and D all wrote their values to the reload area upon archive (byReference=true) and reloaded the new information (reload=true) into A or B.

## 18.2 The Logbook Inquiry Configuration File

---

The Logbook inquiry configuration file governs the fields on which you can query when using the inquiry feature described in section 8.2 *Inquiries*.

### 18.2.1 DTD File Listing for Inquiries

```
<?xml encoding="US-ASCII"?>
<!ELEMENT LogBookReports (filter*)>
<!ELEMENT filter EMPTY>
<!ATTLIST filter
      class CDATA #REQUIRED
      name CDATA #REQUIRED
      type CDATA #REQUIRED>
```

### 18.2.2 Default XML File Listing for Inquiries

Notice that the above DTD file is now (as of v1\_7\_05) included in the XML between the square brackets [ ] on the DOCTYPE line (minus the initial line of the DTD which is no longer needed, since there's one for the XML file itself):

```
<?xml version="1.0"?>
<!DOCTYPE LogBookInquiryConfig [<!ELEMENT LogBookReports (filter*)>
<!ELEMENT filter EMPTY>
<!ATTLIST filter
      class CDATA #REQUIRED
      name CDATA #REQUIRED
      type CDATA #REQUIRED>
]>
<LogBookReports >

<filter type="entry" name="Operators" class="logbook.logdbms.FilterOperator"/>
<filter type="entry" name="Entry Type" class="logbook.logdbms.FilterEntryType"/>
<filter type="entry" name="Date/Time" class="logbook.logdbms.FilterDate"/>
<filter type="entry" name="Keywords" class="logbook.logdbms.FilterKeyword"/>
<filter type="entry" name="Category" class="logbook.logdbms.FilterCategory"/>
<filter type="entry" name="Word Search" class="logbook.logdbms.FilterWordSearch"/>

</LogBookReports >
```

Notice that the names in the above code (e.g., name= "Operators" ) match the search categories in the **INQUIRIES** window:

