Barista Error Handling

A Guide for Barista Application Developers

Introduction

The error handling process in Barista® has been enhanced for flexibility in terms of your client connection types and the level of in-house technical expertise available for troubleshooting. You can configure your BBj/Barista environment to address how issues are reported and how easy or difficult it is to break into code for real time troubleshooting and resolution.

Error Handling Levels

There are three levels of error handling available. Choose the level that suits your needs, depending on such factors as whether this is a production, demo, or development system, and whether or not regular users or perhaps only power users should have access to the console.

- Strict: No console access is allowed. Users may elect to *Send Error Report* to a designated support contact, and then the user will either need to *Abort* the current process or attempt a *Retry* in the case of a file/record lock condition.
- Authorized: Console access is permitted, but only if a password is supplied. Along with the options to *Send Error Report*, *Abort*, or *Retry*, a *Debug* option appears when in this mode.
- Open: Console access is granted any time the *Debug* option is selected. No additional password is required.

Configuration

Both BBj and Barista contain options you can set to configure the desired error handling level.

Strict

The default behavior in BBj is to disallow console access from secure thin client connections. Therefore Strict error handling is the level that's in effect if you make no configuration changes. If a user encounters an unanticipated error, Barista provides information such as the error number, program name, and error line, as seen in Figure 1. If the error is a file or record lock, the error message dialog includes a *Retry* button. Otherwise, the user can *Send Error Report* to the designated support contact, and then *Abort* the process.

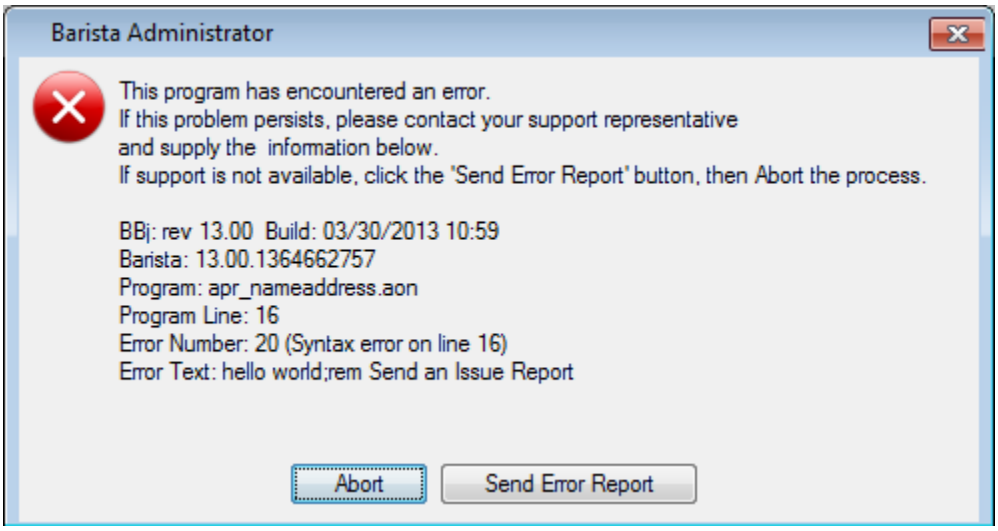

Figure 1: Error message when using Strict error handling

Authorized

If you want to permit access to the console in a controlled fashion, you must configure BBj to allow console access, and also activate settings in barista.cfg to define the prompt used when console access is attempted, along with the required password.

Configure BBj for console access in Enterprise Manager by un-checking the "Disallow Console" box, as shown in Figures 2 and 2a.

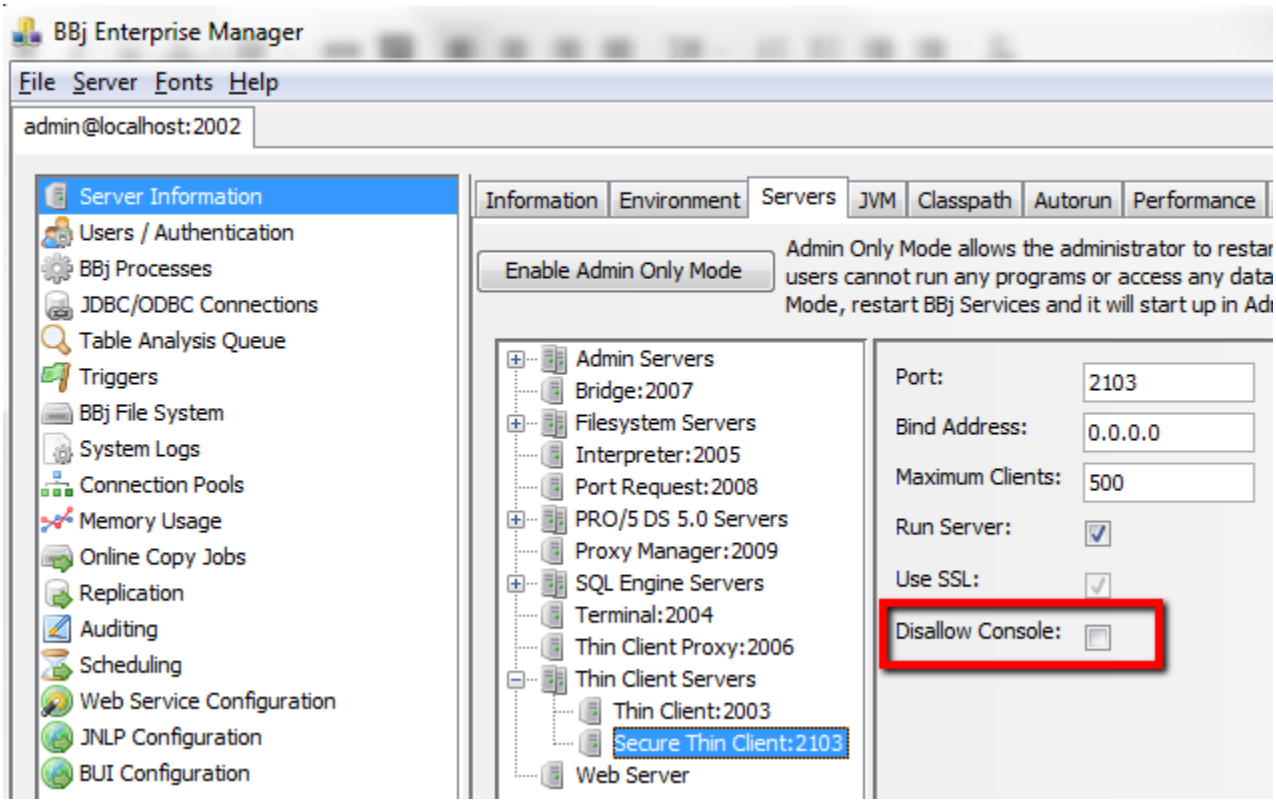

Figure 2: Enterprise Manager Secure Thin Client properties

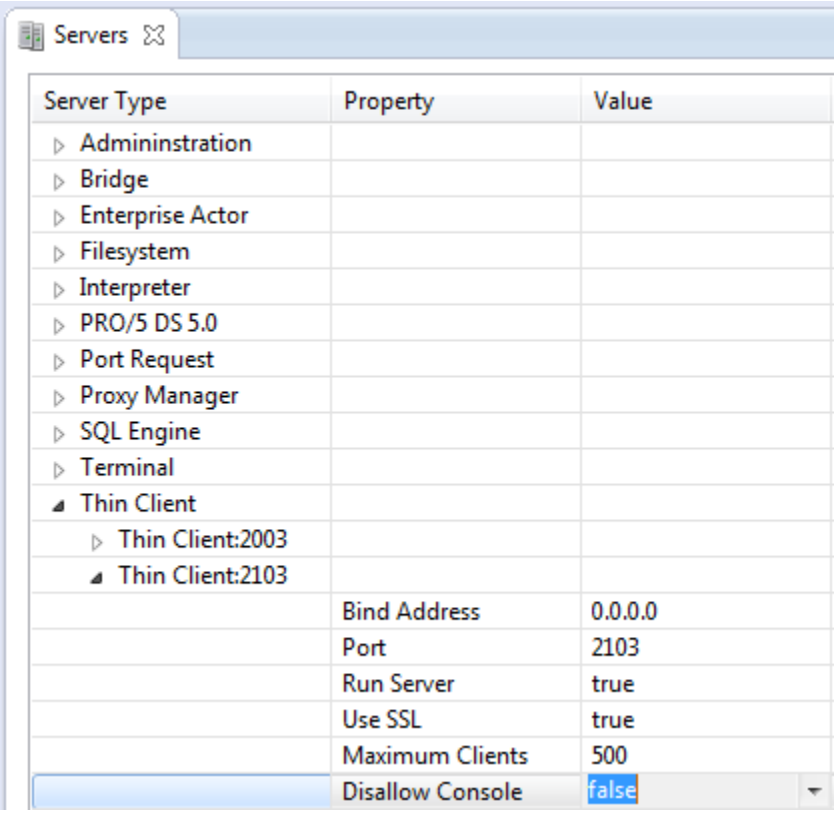

Figure 2a: Eclipse Enterprise Manager Secure Thin Client properties

Next, remove the # characters from the beginning of these three lines in your barista.cfg file, located in Barista's sys/config/<language>/ directory:

```
#set !CONEXIT=true
#set !CONMESS=Please provide the password for console access, or
<enter> to retry:
#set !CONPASS=admin123
```

Given these settings, if a user encounters an unanticipated error, then the error message dialog includes a *Debug* button in addition to the other buttons (Figure 3). If the user clicks *Debug,* a syswindow appears along with the message/prompt specified by the !CONMESS global. Figure 4 shows that the actual `READY>` prompt only appears when the password specified by !CONPASS is supplied. Note that typing `release` aborts the process, although this information is omitted from the default !CONMESS so the user doesn't abort without giving the support contact a chance to troubleshoot the issue.

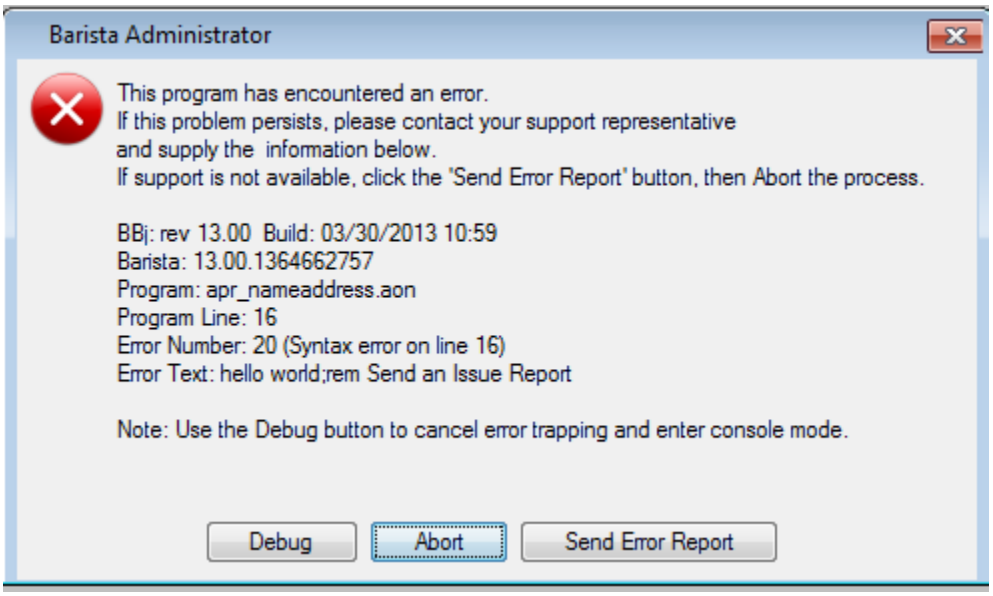

Figure 3: Authorized error handling includes Debug option

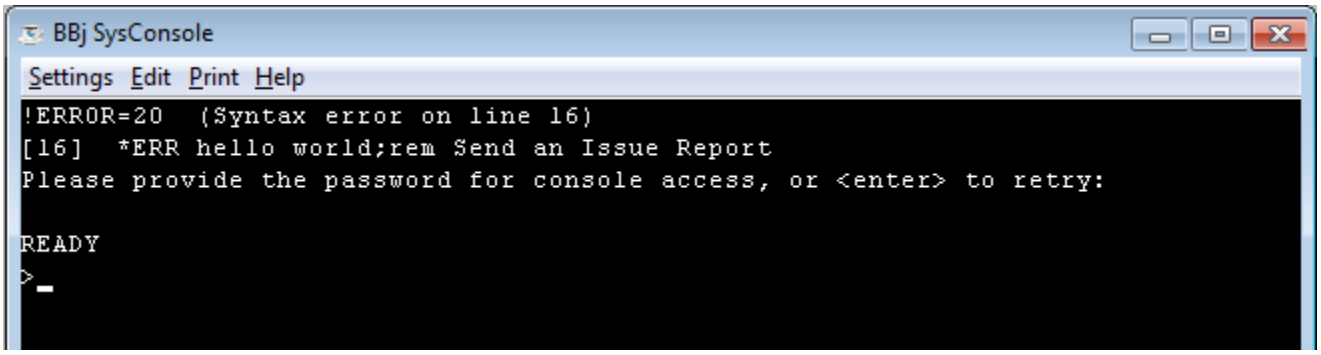

Figure 4: With Authorized error handling console access is only granted after supplying !CONPASS

If you want to change the !CONMESS or !CONPASS for your application, you can either make the change directly in barista.cfg, or (recommended) add these settings to your application's .syn file so they will persist through upgrades.

Open

Open error handling means that console access is permitted to anyone who encounters an unanticipated error, with no additional prompt or password required. This is not the recommended level of error handling for a production environment, but may be fine during periods of development or testing. For Open error handling, all you need to do is configure BBj for console access in Enterprise Manager by un-checking the "Disallow Console" box, as shown in Figure 2.

Similar to Authorized error handling, with Open error handling, if a user encounters an unanticipated error, the error message dialog includes a *Debug* button in addition to the other

buttons. However *unlike* Authorized error handling, if the user clicks *Debug,* a syswindow appears along with the `READY>` prompt; no further authorization is required.

Using the Error Handler

Application programs can make use of the Barista Error Handler with code like the sample below. This sample tests tcb(2), which returns a 0 if the program code is unprotected, and sends the text from the error line into the handler. Inside the handler, if console access is allowed and the error line text isn't empty, then the error message dialog will include a *Debug* button. Barista sets the "ESCAPE" and "RETRY" return values if the user has clicked the *Debug* or *Retry* buttons, respectively. Otherwise the program exits or releases depending on if it has been CALLed or not.

```
seterr std_error
<program code>
std_error: rem --- Standard error handler

    rd_err_text$=""
    if tcb(2)=0 and tcb(5) then rd_err_text$=pgm(tcb(5),tcb(13),err=*next)

call stbl("+DIR_SYP")+"bac_error.bbj",pgm(-2),str(tcb(5)),str(err),rd_err_text$,rd_err_act$
    if pos("ESCAPE"=rd_err_act$) seterr 0; setesc 0
    if pos("RETRY"=rd_err_act$) retry
    if pgm(-1)<>pgm(-2) status=999; exit
    release
```

Sending Issue Reports

Any time an unanticipated error occurs, a user can send a report to their support contact. Pressing the *Send Error Report* button launches a new window (Figure 5) with a *Comments* box where the user can supply additional information.

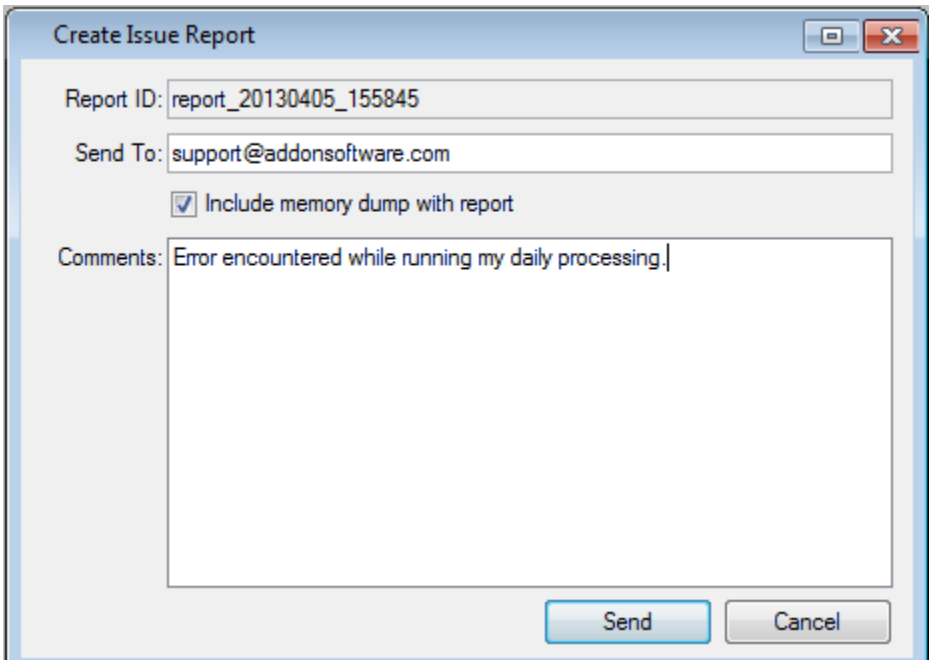

Figure 5: Send an error report with optional memory dump

The sending and recipient email addresses for error reports can be configured in your application's .syn file via the +ISSUE_MAIL_FROM and +ISSUE_MAIL_TO globals, respectively. Alternatively, these settings can be accessed and edited from the Document Processing Monitor's *Configure* button as shown in Figure 6.

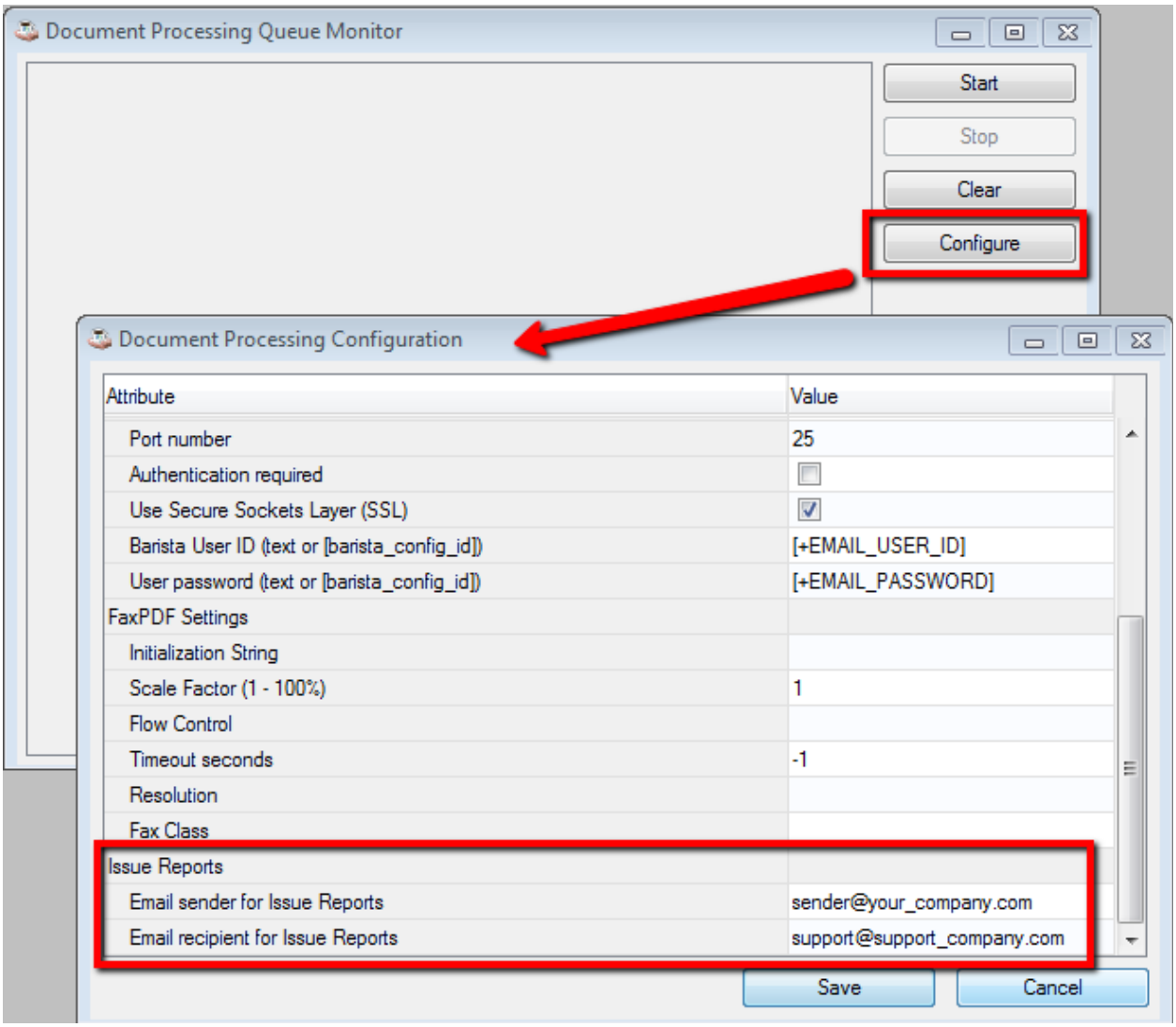

Figure 6: Document Processing Queue Monitor's Configuration grid shows issue report sender and receiver.

Depending on system configuration, the user may be able to select the *Include memory dump with report* checkbox. Set the +ISSUE_DUMP global in barista.cfg or your application's .syn file as follows:

- YES sets the checkbox on by default; users can toggle the selection
- NO sets the checkbox off by default; users can toggle the selection
- ALWAYS sets the checkbox on by default; control is disabled
- NEVER sets the checkbox off by default; control is disabled
- AUTH sets the checkbox off by default; users can toggle the selection if the configured password is supplied

When using the AUTH setting, Barista displays a message that a password is required (Figure 7). Use <alt-enter> to begin password entry. Barista validates the password against

the ISSUE_AUTH record in the Configuration Records table. If the password is correct, the *Include memory dump with report* box can be checked. If incorrect, or if the user dismisses the message, the box is left unchecked.

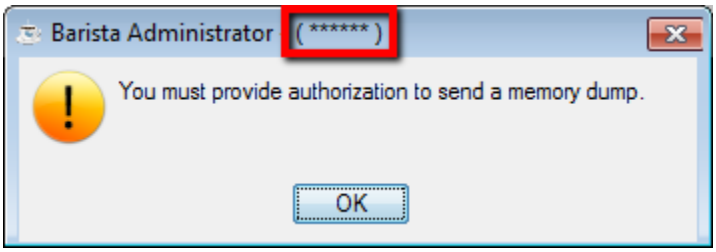

Figure 7: Barista can be configured to prompt for a password before allowing a memory dump.

> *Note: A memory dump contains the current value of all variables, and as such could contain sensitive information.*

When the user clicks the *Send* button, the error information is attached to an email as either a .txt (no memory dump, see Figure 8) or .zip (memory dump selected) file, and placed in Barista's Document Processing Queue. The user sees confirmation that the error report has been placed in the queue, and additional warnings if the Document Processor is not yet configured to send email, or is idle. See Sending Email via the Barista Document Processing Queue for information on configuring and using the document processor. Note that if you use the [+EMAIL_USER_ID] and [+EMAIL_PASSWORD] tokens to configure email, and you do not check the "Load into STBL on execution" button when creating the configuration entries, then the email user ID and password will not be exposed should you send a dump file with the error report.

```
Issue Reporting File
Barista Appplication Framework.  Copyright BASIS International Ltd.
File Name: workarea/report_20130405_162044

BBJ_REV=REV 13.00
BBJ_BUILD=Build: 03/30/2013 10:59
BBJ_SSN=BBX000080

BARISTA_REV=13.00.1364662757

PROGRAM=apr_nameaddress.aon
STMT_NO=16
STMT_TEXT=hello world;rem Send an Issue Report
ERROR_NO=20 (Statement syntax error)
COMMENTS=Error encountered while running my daily processing.


DATE/TIME=2013/04/05 at 16:20:44
USER ID=ADMIN
INFO(3,4)=babaloo
```

Figure 8: Sample error report sent via Barista Document Processor