



LegacyJ PERCobol Exception Processing

Revision April, 2005

The contents of this manual may be revised without prior notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical for any purpose without the expressed written permission of LegacyJ Corporation

Third Edition: April, 2005

LegacyJ has made every effort to ensure that this manual is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time.

© Copyright LegacyJ Corporation 1998, 2005. All Rights Reserved

Preface

This document is a reference for the exception processing and may be valuable in producing mixed COBOL and Java Applications.

LegacyJ PERCobol reserves the right to modify or change this interface without notice.

Contents

PERCOBOL EXCEPTION PROCESSING	1
PROPAGATE	2
RAISE VERB	3
EXIT RAISING.....	4
GOBACK RAISING.....	5
USE VERB	6
EXCEPTION NAMES.....	7
JAVA LANGUAGE INTEGRATION NOTES	10

PERCobol Exception Processing

This documentation addendum is currently aimed mostly at allowing Java integration but it does cover the COBOL usage.

There is also an additional way of catching exceptions which can be done, the TRY/CATCH mechanism, which is handy for localized handling. (The CATCH has an optional PIC X(n) identifier which captures the text of the exception.) Note, that if an exception is not caught by the COBOL code it may propagate either to a calling Java program or to a general exception printing routine (for the top-level program).

The only error generally possible with sort is an OutOfMemory, so the TRY may be sufficient to wrap around the SORT.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. thrower.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 ID-1 PIC X(40).  
  
PROCEDURE DIVISION.  
MAIN-PARA.  
  
    DISPLAY "begin" upon sysout  
  
    TRY  
        RAISE TEXT "Error"  
    CATCH ID-1  
        DISPLAY "Caught: " ID-1 UPON SYSOUT  
    END-TRY  
  
    DISPLAY "end" upon sysout  
    .
```

PROPAGATE

The >>PROPAGATE ON compiler instruction turns on the ability to propagate exceptions up the call chain. The >>PROPAGATE OFF compiler instruction disables this ability. OFF is the default. (This compiler instruction, including the >>, must occur in program area A or B.)

For instance, a CALL without ON EXCEPTION clause to a non-existent program will display an error dialog if propagate is off. With propagate on, the EC-PROGRAM exception condition will continue propagating to the USE clause (if present) or the next-level up in the call chain.

The RAISE and RAISING verbs raise exceptions.

RAISE Verb

```
RAISE LAST EXCEPTION  
RAISE object-reference  
RAISE EXCEPTION exception-name  
RAISE TEXT text-name  
RAISE CLASS class-name
```

EXIT RAISING

```
EXIT PROGRAM RAISING LAST EXCEPTION
EXIT PROGRAM RAISING object-reference
EXIT PROGRAM RAISING EXCEPTION exception-name
EXIT PROGRAM RAISING TEXT text-name
EXIT PROGRAM RAISING CLASS class-name
```

GOBACK RAISING

```
GOBACK RAISING LAST EXCEPTION
GOBACK RAISING object-reference
GOBACK RAISING EXCEPTION exception-name
GOBACK RAISING TEXT text-name
GOBACK RAISING CLASS class-name
```

To raise an EC-BOUND exception, for instance, the code would be:

```
RAISE EXCEPTION EC-BOUND
```

and to raise a user exception testing:

```
RAISE EXCEPTION EC-USER-TESTING
```

An example of raising a java.rmi.RemoteException:

```
77 OBJECT REFERENCE C"java.rmi.RemoteException".
INVOKE MY-REMOTE USING BY VALUE "My Remote Exception" GIVING MY-REMOTE
RAISE MY-REMOTE
```

The EXIT PROGRAM RAISING will only raise an exception where an EXIT PROGRAM would have an effect. The GOBACK RAISING will perform a GOBACK after the exception is handled.

USE Verb

Format 3 (exception-name)

```
USE [AFTER] {EXCEPTION OBJECT | EO} class-name-1.
```

Format 4 (exception-object)

```
USE [AFTER] {EXCEPTION CONDITION | EC} exception-name-1 ...
```

The USE verb of format 3 and format 4 catches exceptions raised by RAISE or a RAISING clause. The section executes as its handler. If the exception is considered fatal, then a STOP RUN exception occurs after execution of the USE section.

Within the bounds of such a format 4 USE section, the following functions are available:

```
FUNCTION EXCEPTION-FILE
FUNCTION EXCEPTION-LOCATION
FUNCTION EXCEPTION-STATEMENT
FUNCTION EXCEPTION-STATUS
```

And an example of catching a java.rmi.RemoteException:

```
DECLARATIVES.
REMOTE-H SECTION.
    USE EXCEPTION OBJECT "java.rmi.RemoteException".
REMOTE-P.
    DISPLAY "Got remote exception" UPON SYSOUT
.
```

Exception Names

The following exception names are available. The classname is also listed, necessary only if catching the exception name from Java.

Exception Name	Classname
EC-ALL	com.legacyj.exception.EcAll
EC-ARGUMENT	com.legacyj.exception.EcArgument
EC-ARGUMENT-FUNCTION	com.legacyj.exception.EcArgumentFunction
EC-ARGUMENT-IMP	com.legacyj.exception.EcArgumentImp
EC-BOUND	com.legacyj.exception.EcBound
EC-BOUND-IMP	com.legacyj.exception.EcBoundImp
EC-BOUND-ODO	com.legacyj.exception.EcBoundOdo
EC-BOUND-PTR	com.legacyj.exception.EcBoundPtr
EC-BOUND-REF-MOD	com.legacyj.exception.EcBoundRefMod
EC-BOUND-SUBSCRIPT	com.legacyj.exception.EcBoundSubscript
EC-DATA	com.legacyj.exception.EcData
EC-DATA-CONVERSION	com.legacyj.exception.EcDataConversion
EC-DATA-IMP	com.legacyj.exception.EcDataImp
EC-DATA-INCOMPATIBLE	com.legacyj.exception.EcDataIncompatible
EC-DATA-PTR-NULL	com.legacyj.exception.EcDataPtrNull
EC-FLOW	com.legacyj.exception.EcFlow
EC-FLOW-GLOBAL-EXIT	com.legacyj.exception.EcFlowGlobalExit
EC-FLOW-GLOBAL-GOBACK	com.legacyj.exception.EcFlowGlobalGoback
EC-FLOW-IMP	com.legacyj.exception.EcFlowImp
EC-FLOW-IMP-STOP-INVALID	com.legacyj.exception.EcFlowImpStopInvalid
EC-FLOW-RELEASE	com.legacyj.exception.EcFlowRelease
EC-FLOW-REPORT	com.legacyj.exception.EcFlowReport
EC-FLOW-RETURN	com.legacyj.exception.EcFlowReturn
EC-FLOW-USE	com.legacyj.exception.EcFlowUse
EC-I-O	com.legacyj.exception.EcIO
EC-I-O-AT-END	com.legacyj.exception.EcIOAtEnd
EC-I-O-EOP	com.legacyj.exception.EcIOEOP
EC-I-O-EOP-OVERFLOW	com.legacyj.exception.EcIOEOPOverflow
EC-I-O-FILE-SHARING	com.legacyj.exception.EcIOFileSharing
EC-I-O-IMP	com.legacyj.exception.EcIOImp
EC-I-O-INVALID-KEY	com.legacyj.exception.EcIOInvalidKey
EC-I-O-LINAGE	com.legacyj.exception.EcIOLinage
EC-I-O-LOGIC-ERROR	com.legacyj.exception.EcIOLogicError
EC-I-O-PERMANENT-ERROR	com.legacyj.exception.EcIOPermanentError
EC-I-O-RECORD-OPERATION	com.legacyj.exception.EcIORecordOperation
EC-IMP	com.legacyj.exception.EcImpImp
EC-LOCALE	com.legacyj.exception.EcLocale
EC-LOCALE-IMP	com.legacyj.exception.EcLocaleImp
EC-LOCALE-INCOMPATIBLE	com.legacyj.exception.EcLocaleIncompatible
EC-LOCALE-INVALID	com.legacyj.exception.EcLocaleInvalid
EC-LOCALE-INVALID-PTR	com.legacyj.exception.EcLocaleInvalidPtr
EC-LOCALE-MISSING	com.legacyj.exception.EcLocaleMissing
EC-LOCALE-SIZE	com.legacyj.exception.EcLocaleSize
EC-OO	com.legacyj.exception.EcOO
EC-OO-CONFORMANCE	com.legacyj.exception.EcOOConformance
EC-OO-EXCEPTION	com.legacyj.exception.EcOOException
EC-OO-IMP	com.legacyj.exception.EcOOImp
EC-OO-METHOD	com.legacyj.exception.EcOOMethod
EC-OO-NULL	com.legacyj.exception.EcOONull
EC-OO-RESOURCE	com.legacyj.exception.EcOOResource
EC-OO-UNIVERSAL	com.legacyj.exception.EcOOUniversal

Exception Name	Classname
EC-ORDER	com.legacyj.exception.EcOrderOrder
EC-ORDER-MAP	com.legacyj.exception.EcOrderMap
EC-ORDER-NOT-SUPPORTED	com.legacyj.exception.EcOrderNotSupported
EC-OVERFLOW	com.legacyj.exception.EcOverflow
EC-OVERFLOW-IMP	com.legacyj.exception.EcOverflowImp
EC-OVERFLOW-STRING	com.legacyj.exception.EcOverflowString
EC-OVERFLOW-UNSTRING	com.legacyj.exception.EcOverflowUnstring
EC-PROGRAM	com.legacyj.exception.EcProgram
EC-PROGRAM-ARG-MISMATCH	com.legacyj.exception.EcProgramArgMismatch
EC-PROGRAM-ARG-OMITTED	com.legacyj.exception.EcProgramArgOmitted
EC-PROGRAM-CANCEL-ACTIVE	com.legacyj.exception.EcProgramCancelActive
EC-PROGRAM-IMP	com.legacyj.exception.EcProgramImp
EC-PROGRAM-NOT-FOUND	com.legacyj.exception.EcProgramNotFound
EC-PROGRAM-PTR-NULL	com.legacyj.exception.EcProgramPtrNull
EC-PROGRAM-RECURSIVE-CALL	com.legacyj.exception.EcProgramRecursiveCall
EC-PROGRAM-RESOURCES	com.legacyj.exception.EcProgramResources
EC-RAISING	com.legacyj.exception.EcRaising
EC-RAISING-IMP	com.legacyj.exception.EcRaisingImp
EC-RAISING-NOT-SPECIFIED	com.legacyj.exception.EcRaisingNotSpecified
EC-RANGE	com.legacyj.exception.EcRange
EC-RANGE-IMP	com.legacyj.exception.EcRangeImp
EC-RANGE-INDEX	com.legacyj.exception.EcRangeIndex
EC-RANGE-INSPECT-SIZE	com.legacyj.exception.EcRangeInspectSize
EC-RANGE-INVALID	com.legacyj.exception.EcRangeInvalid
EC-RANGE-PERFORM-VARYING	com.legacyj.exception.EcRangePerformVarying
EC-RANGE-PTR	com.legacyj.exception.EcRangePtr
EC-RANGE-SEARCH-INDEX	com.legacyj.exception.EcRangeSearchIndex
EC-RANGE-SEARCH-NO-MATCH	com.legacyj.exception.EcRangeSearchNoMatch
EC-REPORT	com.legacyj.exception.EcReport
EC-REPORT-ACTIVE	com.legacyj.exception.EcReportActive
EC-REPORT-COLUMN-OVERLAP	com.legacyj.exception.EcReportColumnOverlap
EC-REPORT-FILE-MODE	com.legacyj.exception.EcReportFileMode
EC-REPORT-IMP	com.legacyj.exception.EcReportImp
EC-REPORT-INACTIVE	com.legacyj.exception.EcReportInactive
EC-REPORT-LINE-OVERLAP	com.legacyj.exception.EcReportLineOverlap
EC-REPORT-NOT-TERMINATED	com.legacyj.exception.EcReportNotTerminated
EC-REPORT-PAGE-LIMIT	com.legacyj.exception.EcReportPageLimit
EC-REPORT-PAGE-WIDTH	com.legacyj.exception.EcReportPageWidth
EC-REPORT-SUM-SIZE	com.legacyj.exception.EcReportSumSize
EC-REPORT-VARYING	com.legacyj.exception.EcReportVarying
EC-SCREEN	com.legacyj.exception.EcScreen
EC-SCREEN-FIELD-OVERLAP	com.legacyj.exception.EcScreenFieldOverlap
EC-SCREEN-IMP	com.legacyj.exception.EcScreenImp
EC-SCREEN-ITEM-TRUNCATED	com.legacyj.exception.EcScreenItemTruncated
EC-SCREEN-LINE-NUMBER	com.legacyj.exception.EcScreenLineNumber
EC-SCREEN-STARTING-COLUMN	com.legacyj.exception.EcScreenStartingColumn
EC-SIZE	com.legacyj.exception.EcSize
EC-SIZE-ADDRESS	com.legacyj.exception.EcSizeAddress
EC-SIZE-EXPONENTIATION	com.legacyj.exception.EcSizeExponentiation
EC-SIZE-IMP	com.legacyj.exception.EcSizeImp
EC-SIZE-OVERFLOW	com.legacyj.exception.EcSizeOverflow
EC-SIZE-TRUNCATION	com.legacyj.exception.EcSizeTruncation
EC-SIZE-UNDERFLOW	com.legacyj.exception.EcSizeUnderflow
EC-SIZE-ZERO-DIVIDE	com.legacyj.exception.EcSizeZeroDivide
EC-SORT-MERGE	com.legacyj.exception.EcSortMerge
EC-SORT-MERGE-ACTIVE	com.legacyj.exception.EcSortMergeActive
EC-SORT-MERGE-FILE-OPEN	com.legacyj.exception.EcSortMergeFileOpen
EC-SORT-MERGE-IMP	com.legacyj.exception.EcSortMergeImp

Exception Name	Classname
EC-SORT-MERGE-RELEASE	com.legacyj.exception.EcSortMergeRelease
EC-SORT-MERGE-RETURN	com.legacyj.exception.EcSortMergeReturn
EC-SORT-MERGE-SEQUENCE	com.legacyj.exception.EcSortMergeSequence
EC-STORAGE	com.legacyj.exception.EcStorage
EC-STORAGE-IMP	com.legacyj.exception.EcStorageImp
EC-STORAGE-NOT-ALLOC	com.legacyj.exception.EcStorageNotAlloc
EC-STORAGE-NOT-AVAIL	com.legacyj.exception.EcStorageNotAvail
EC-USER	com.legacyj.exception.EcUser
EC-VALIDATE	com.legacyj.exception.EcValidate
EC-VALIDATE-CONTENT	com.legacyj.exception.EcValidateContent
EC-VALIDATE-FORMAT	com.legacyj.exception.EcValidateFormat
EC-VALIDATE-IMP	com.legacyj.exception.EcValidateImp
EC-VALIDATE-RELATION	com.legacyj.exception.EcValidateRelation
EC-VALIDATE-VARYING	com.legacyj.exception.EcValidateVarying

Java language integration notes

Objects may be raised directly. Raising a `java.lang.Throwable` extender will issue the exact exception to Java; raising other objects will issue `com.legacyj.exception.EcObject` exceptions. Any checked exceptions (those not extending `RuntimeException` or `Error`) must be declared in the `THROWS` clause of the `PROGRAM-ID`, or they will be wrapped in `EcObject`.

Catching `com.legacyj.exception.EcAll` will catch all exception conditions and wrapped objects (`com.legacyj.exception.EcObject`), but will not catch declared throwable exceptions. These exceptions need to be caught using the same name declared, as would be expected in the Java language.