Oracle Tuxedo Application Runtime for Batch
Reference Guide
12c Release 1 (12.1.1)

September 2013
Contents

Introduction

Z/OS JCL in the Batch Runtime Environment

Introduction to Z/OS JCL in the Batch Runtime Environment .......................... 2-1
z/OS JCL Cards in the Batch Runtime Environment ........................................ 2-1
  JCL Card Equivalence Table ............................................................... 2-2
  DCB Parameter of DD Statement ......................................................... 2-13
  SYSOUT Parameter of DD Statement .................................................... 2-15
  DATACLAS Parameter of DD Statement ............................................... 2-15
  RESTART Parameter of JOB Statement ................................................. 2-15
  In-Stream File ..................................................................................... 2-15
  Concatenation of Files .......................................................................... 2-16
  Override of Files .................................................................................. 2-16
  Execution of External Sysin .................................................................... 2-16
  General Utility Commands Equivalence Table ........................................ 2-16
  Sort Utilities ....................................................................................... 2-20
  Sort Utilities Equivalence Tables ......................................................... 2-20
  DB2 Utilities ....................................................................................... 2-21

Oracle Tuxedo Application Runtime for Batch Functions

Introduction to the Batch Runtime Commands .............................................. 3-1
  Emulating z/OS JCL Logic and Architecture ............................................ 3-1
  EJR Syntax ......................................................................................... 3-3
Tools for Managing the Execution of Jobs ........................................ 3-4
Oracle Tuxedo Application Runtime for Batch Functions .................. 3-8
Naming Convention ................................................................. 3-8
Reference Page Command Syntax .............................................. 3-8
Reference .................................................................................. 3-10

**Tuxedo Job Enqueueing Service (TuxJES)**

genjesprofile ............................................................................. 4-2
artjesadmin ................................................................................. 4-2
artjescleanlock ......................................................................... 4-8
ARTJESADM ............................................................................. 4-9
ARTJESCONV ........................................................................... 4-11
ARTJESINITIATOR ................................................................. 4-11
ARTJESPURGE ......................................................................... 4-13
TuxJES Queue System ............................................................... 4-13
   The TuxJES Queue Creation Script ........................................ 4-14
   Recommended /Q Creation Values ........................................... 4-14
TuxJES User Substitution ............................................................ 4-15
   Terms .................................................................................. 4-15
   Job Owner Designation in Different Scenarios ....................... 4-15
   Permissions of Job Operations .............................................. 4-16
   Enabling User Substitution in TuxJES .................................... 4-17
The Oracle Tuxedo Application Runtime for Batch normalizes Korn shell script formats by proposing a script model in which the different execution phases are clearly identified, and provides the Tuxedo Job Enqueueing Service (TuxJES), which emulates the major functions of Mainframe JES2.

This guide consists of three main parts:

- The first part describes the equivalencies that exist between JCL cards, general utility commands and sorts on the one hand and the Batch Runtime functions on the other.

- The second part describes how the Korn shell scripts are structured to emulate a JCL type processing of jobs. The different functions of the Batch Runtime that are used in these scripts are then described in detail.

- The third part describes the servers and utilities for TuxJES.
Introduction
This chapter contains the following topics:

- Introduction to z/OS JCL in the Batch Runtime Environment
- z/OS JCL Cards in the Batch Runtime Environment
- Sort Utilities
- DB2 Utilities

**Introduction to z/OS JCL in the Batch Runtime Environment**

This section describes how to find equivalents for z/OS JCL statements in the target environment. Some of these equivalents point to the Batch Runtime functions, other equivalents may rely directly on UNIX or Tuxedo features. In some cases, there may be no equivalent and a work-around solution may be necessary.

It is not the purpose of this document to describe z/OS JCL, for any explanation of JCL statements, please see the z/OS Internet Library.

**z/OS JCL Cards in the Batch Runtime Environment**

The following tables lists the JCL card parameters and the related command in the Batch Runtime:
In the column “status”, the following abbreviations are used:

N.R. means “not relevant”
N.S. means “not supported”

### JCL Card Equivalence Table

Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>/**</td>
<td></td>
<td>Comments</td>
<td>supported</td>
</tr>
<tr>
<td>*/</td>
<td></td>
<td>In-stream data delimiter</td>
<td>supported</td>
</tr>
<tr>
<td>//</td>
<td></td>
<td>End of job</td>
<td>supported</td>
</tr>
<tr>
<td>COMMAND</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>CNTL/ENDCNTL</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
</tbody>
</table>
### Table 2-1: JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>*</td>
<td>m_FileAssign -i</td>
<td>supported</td>
</tr>
<tr>
<td>ACCODE</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>AMP</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>AVGREC</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BLKSIZE</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BLKSIZE2</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>CCSD</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>CHKPT</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>CNTL</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>DATA</td>
<td></td>
<td>m_FileAssign</td>
<td>supported</td>
</tr>
<tr>
<td>DATACLAS</td>
<td></td>
<td>see DATACLAS chapter</td>
<td>supported</td>
</tr>
<tr>
<td>DCB</td>
<td></td>
<td>see DCB chapter</td>
<td>supported</td>
</tr>
<tr>
<td>DISP</td>
<td>m_FileAssign -d &lt;DISP option&gt;</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td>DLM</td>
<td>m_FileAssign -D &lt;delimiter&gt;</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td>DSID</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>DSNAME</td>
<td>m_FileAssign</td>
<td>supported</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>DSNTYPE</td>
<td>m_FileAssign with /dev/null</td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DUMMY</td>
<td>m_FileAssign with /dev/null</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DYNAM</td>
<td>m_FileAssign with /dev/null</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>m_FileAssign -e</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>FILEDATA</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>FREE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>KEYLEN</td>
<td>m_FileAssign -k</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>KEYOFF</td>
<td>m_FileAssign -k</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>LABEL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>LGSTREAM</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>m_FileAssign -s</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>LRECL</td>
<td>m_FileAssign -r</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>MSVGP</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PATH</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PATHDISP</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PATHMODE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PATHOPTS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PROTECT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>QName</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>RECFM</td>
<td>m_FileAssign -r</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>RECORG</td>
<td>m_FileAssign -t</td>
<td>supported</td>
</tr>
<tr>
<td>JCL Card</td>
<td>Parameter</td>
<td>Equivalent in Target Environment</td>
<td>Status</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>DD (continued)</td>
<td>REFDD</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>m_FileAssign -e</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>RLS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SECMODEL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SUBSYS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>TERM</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>VOLUME</td>
<td>m_FileAssgin -v</td>
<td>supported</td>
</tr>
</tbody>
</table>
### Table 2-1 JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD : Printing parameters</td>
<td>BURST</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CCSID</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>CHARS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>COPIES</td>
<td>m_OutputAssign -c</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DEST</td>
<td>m_OutputAssign -d</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>FCB</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FLASH</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>HOLD</td>
<td>m_OutputAssign -H</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>MODIFY</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>OUTLIM</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>m_OutputAssign -o</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SPIN</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSOUT</td>
<td>see SYSOUT chapter</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>UCS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>DD: Special statements</td>
<td>JOBLIB</td>
<td>m_JobLibset</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>STEPLIB</td>
<td>m_StepLibset</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SYSABEND</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSMDUMP</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSUDUMP</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSCHK</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSIN</td>
<td>m_FileAssign</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SYSCKEEOV</td>
<td></td>
<td>N.S.</td>
</tr>
</tbody>
</table>
Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>ACCT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>ADDRSPC</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>CCSID</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>COND</td>
<td>m_CondExec</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DPRTY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DYNAMNBR</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>MEMLIMIT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PARM</td>
<td>m_ProgramExec</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>PERFORM</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>PGM</td>
<td>m_ProgramExec</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>PROC</td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>RD</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>REGION</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>RLSTMOUT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>TIME</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>IF THEN ELSE END</td>
<td>m_CondIf, m_CondElse, m_CondEndif</td>
<td>supported</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>MEMBER</td>
<td>m_ShellInclude</td>
<td></td>
</tr>
<tr>
<td>JCLLIB</td>
<td>ORDER</td>
<td>m_JclLibSet</td>
<td></td>
</tr>
</tbody>
</table>
Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>&lt;jobname&gt;</td>
<td>m_JobBegin</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>&lt;accounting&gt;</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>&lt;prog. name&gt;</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>ADDRSPC</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BYTES</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>CARDS</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>CCSID</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>CLASS</td>
<td></td>
<td>m_JobBegin -c (with TuxJES)</td>
<td>supported</td>
</tr>
<tr>
<td>COND</td>
<td></td>
<td>m_JobBegin -C</td>
<td>supported</td>
</tr>
<tr>
<td>GROUP</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>JESLOG</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>LINES</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>MEMLIMIT</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>MSGCLASS</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>NOTIFY</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>PAGES</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>PASSWORD</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>PERFORM</td>
<td></td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>PRTY</td>
<td></td>
<td>m_JobBegin -p (with TuxJES)</td>
<td>supported</td>
</tr>
<tr>
<td>RD</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
</tbody>
</table>
Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB (continued)</td>
<td>SECLABEL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SCHENV</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>TIME</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>TYPRUN</td>
<td>m_JobBegin -t (with TuxJES)</td>
<td>supported</td>
</tr>
<tr>
<td>USER</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>ADDRESS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>AFPSTATS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>BUILDING</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>BURST</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CHARS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CKPTLINE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CKPTPAGE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CKPTSEC</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CLASS</td>
<td>m_OutputSet -c</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>COLORMAP</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>COMPACT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>COMSETUP</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>COPIES</td>
<td>m_OutputSet -n</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DATAACK</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEFAULT</td>
<td>m_OutputSet -D</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DEPT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEST</td>
<td>m_OutputSet -d</td>
<td>supported</td>
</tr>
</tbody>
</table>
Table 2-1 JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td>DPAGELBL</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>(continued)</td>
<td>DUPLEX</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FCB</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FLASH</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FORMDEF</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FORMLEN</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>FORMS</td>
<td>m_OutputSet -f</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>FSSDATA</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>GROUPID</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>INDEX</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>INTRAY</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>JESDS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>LINDEX</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>LINECT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILBCC</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILCC</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILCC</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILFILE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILFROM</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAILTO</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MODIFY</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>NAME</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>NOTIFY</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>OFFSETXB</td>
<td></td>
<td>N.S.</td>
</tr>
</tbody>
</table>
### Table 2-1  JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td>OFFSETXF</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OFFSETYB</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OFFSETYF</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTBIN</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OUTDISP</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVERLAYB</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVERLAYF</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVFL</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAGDEEF</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIMSG</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PORTNO</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRMODE</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRTATTRS</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRERROR</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRTOPTNS</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRTQUEUE</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRTY</td>
<td>m_OutputSet -p supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPLYTO</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RESPMT</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RETAINF</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RETAINS</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RETRYL</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RETRYT</td>
<td>N.S.</td>
<td></td>
</tr>
</tbody>
</table>
Table 2-1 JCL Card Equivalences

<table>
<thead>
<tr>
<th>JCL Card</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT (continued)</td>
<td>ROOM</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>SYSAREA</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>THRESHLD</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>TITLE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>TRC</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>UCS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>USERDATA</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>USERLIB</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>USERPATH</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>WRITER</td>
<td>m_OutputSet -w</td>
<td>supported</td>
</tr>
<tr>
<td>PROC</td>
<td></td>
<td>m_ProcInclude</td>
<td>supported</td>
</tr>
<tr>
<td>in-stream PROC</td>
<td></td>
<td>m_ProcBegin</td>
<td>supported</td>
</tr>
<tr>
<td>in-stream PEND</td>
<td></td>
<td>m_ProcEnd</td>
<td>supported</td>
</tr>
<tr>
<td>SET</td>
<td></td>
<td>m_SymbolSet</td>
<td>supported</td>
</tr>
<tr>
<td>XMIT (Records)</td>
<td></td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td>XMIT (JOB)</td>
<td></td>
<td>m_SetJobExecLocation</td>
<td>supported</td>
</tr>
<tr>
<td>XEQ</td>
<td></td>
<td>m_SetJobExecLocation</td>
<td>supported</td>
</tr>
<tr>
<td>ROUTE</td>
<td>PRINT</td>
<td>Share with NFS</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>PUNCH</td>
<td>Share with NFS</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>XEQ</td>
<td>m_SetJobExecLocation</td>
<td>supported</td>
</tr>
</tbody>
</table>
## DCB Parameter of DD Statement

<table>
<thead>
<tr>
<th>DCB parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFANL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BFTEK</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BLKSIZE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFIN</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFMAX</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFNO</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFOFF</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFOUT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>BUFSIZE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>CPRI</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>CYLOFL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>DEN</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>DIAGNS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>DSORG</td>
<td>m_FileAssign -t</td>
<td>supported</td>
</tr>
<tr>
<td>EROPT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>FUNC</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>GNCP</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>INTVL</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>IPLTXID</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>KEYLEN</td>
<td>m_FileAssign -k</td>
<td>supported</td>
</tr>
</tbody>
</table>
### Table 2-2 DCB Parameter of DD Statement

<table>
<thead>
<tr>
<th>DCB parameter</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMCT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>LRECL</td>
<td>m_FileAssign -r</td>
<td>supported</td>
</tr>
<tr>
<td>MODE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>NCP</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>NTM</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>OPTCD</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>PCI</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>PRTSP</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>RECFM</td>
<td>m_FileAssign -r</td>
<td>supported</td>
</tr>
<tr>
<td>RESERVE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>RKP</td>
<td>m_FileAssign -t</td>
<td>supported</td>
</tr>
<tr>
<td>STACK</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>TRESH</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td>TRTCH</td>
<td></td>
<td>N.R.</td>
</tr>
</tbody>
</table>
SYSOUT Parameter of DD Statement

Table 2-3  SYSOUT Parameter of DD Statement

<table>
<thead>
<tr>
<th>SYSOUT parameter</th>
<th>Usage</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>First parameter</td>
<td>Class</td>
<td>m_OutputAssign -c</td>
<td>supported</td>
</tr>
<tr>
<td>Second parameter</td>
<td>INTRDR</td>
<td>m_OutputAssign -w</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SMTP</td>
<td>m_OutputAssign -w</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>&lt;writer&gt;</td>
<td>m_OutputAssign -w</td>
<td>supported</td>
</tr>
<tr>
<td>Third parameter</td>
<td>Form name</td>
<td>m_OutputAssign -f</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>Code name</td>
<td></td>
<td>N.S.</td>
</tr>
</tbody>
</table>

DATACLAS Parameter of DD Statement

The DATACLAS parameter is taken into account at WB JCL level only.

Using the DATACLASS informations, extracted from z/OS, the WB JCL updates the m_FileAssign function (or m_FileBuild from DATACLAS parameter in IDCAMS DEFINE commands).

RESTART Parameter of JOB Statement

Only the Deferred Step Restart type is managed.

The other types, Automatic Restart (Step or Checkpoint) and Deferred Checkpoint Restart, are not managed.

A deferred step restart of a job is caused by coding the function m_JobBegin with the -r parameter containing a job step and by submitting the job again.

In-Stream File

The in-stream files are supported using the m_FileAssign function with the -i parameter.
Concatenation of Files
The concatenation of files (or in-stream files) is supported using the `m_FileAssign` function with the `-C` parameter.

Override of Files
The overriding feature is supported.
Use the `m_FileOverride` function for the files managed by the `m_FileAssign` function.
Use the `m_OutputOverride` function for the files managed by the `m_OutputAssign` function.

Execution of External Sysin
Use the `m_UtilityExec` function in order to execute commands stored in a file.

Network Job Entry (NJE)
The following table describes the supported JES2 control statements and parameters in Oracle Tuxedo Application Rehosting Workbench and Batch Runtime.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Supported Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>/*XEQ</td>
<td>/*XEQ nodename</td>
</tr>
<tr>
<td>/*ROUTE</td>
<td>/*ROUTE PRINT LOCAL</td>
</tr>
<tr>
<td></td>
<td>/*ROUTE PRINT nodename</td>
</tr>
<tr>
<td></td>
<td>/*ROUTE PUNCH nodename</td>
</tr>
<tr>
<td></td>
<td>/*ROUTE XEQ nodename</td>
</tr>
<tr>
<td>/*XMIT</td>
<td>/*XMIT nodename</td>
</tr>
</tbody>
</table>

ABEND Routine
The implementation of `ILBOABN0` is different between MicroFocus and COBOL-IT.
Microfocus provides a mechanism to register an exit routine, which can be automatically called when the ABEND routine calls `stop run. stop run` terminates the process after the registered callback function completes cleaning.
COBOL-IT does not provide such mechanism. In the ABEND routine for COBOL-IT, the function mentioned above is called explicitly, stop run is called, and then the process is terminated directly at ILBOABN0.

The source of ILBOABN0.cbl is under Batch_RT/ejr/SAMPLE directory.

### General Utility Commands Equivalence Table

<table>
<thead>
<tr>
<th>Utility</th>
<th>Command</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDCAMS (modal commands)</td>
<td>IF/THEN/ELSE</td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SET</td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>CANCEL</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DO/END</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>MAXCC</td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>LASTCC</td>
<td></td>
<td>supported</td>
</tr>
<tr>
<td>Utility</td>
<td>Command</td>
<td>Equivalent in Target Environment</td>
<td>Status</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
<td>-----------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>IDCAMS</td>
<td>ALLOCATE (main parameters)</td>
<td>m_FileBuild</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>ALTER NEWNAME</td>
<td>m_FileRename</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Renaming for VSAM RDB files is not supported.</td>
</tr>
<tr>
<td></td>
<td>ALTER LIBRARYENTRY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>ALTER VOLUMEENTRY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>BLDINDEX</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>CREATE LIBRARYENTRY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>CREATE VOLUMEENTRY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DCOLLECT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEFINE ALIAS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEFINE AIX</td>
<td>m_FileBuild -K</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DEFINE CLUSTER</td>
<td>m_FileBuild</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DEFINE GDG</td>
<td>m_GenDefine</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DEFINE NONVSAM</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEFINE PAGESPACE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DEFINE PATH</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DEFINE USRCATALOG</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DELETE ALIAS</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DELETE AIX</td>
<td>m_FileBuild -R</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DELETE CLUSTER</td>
<td>m_FileDelete</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>DELETE GDG</td>
<td>m_FileDelete</td>
<td>supported</td>
</tr>
</tbody>
</table>
### Table 2-5 General Utility Commands Equivalences

<table>
<thead>
<tr>
<th>Utility</th>
<th>Command</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDCAMS (continued)</td>
<td>DELETE NVR</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DELETE PAGESPACE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DELETE PATH</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DELETE TRUENAME</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>DEL USERCATALOG</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DELETE VOLUMEENTRY</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DELETE VVR</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>DIAGNOSE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>EXAMINE</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>EXPORT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>EXPORT DISCONNECT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>IMPORT</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>IMPORT CONNECT</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>LISTCAT</td>
<td>m_FileListcat</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>LISTDATA</td>
<td></td>
<td>N.S.</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
<td>m_FilePrint</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>REPRO</td>
<td>m_FileRepro</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SETCACHE</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>SHCDS</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>VERIFY</td>
<td></td>
<td>N.S.</td>
</tr>
</tbody>
</table>
### Table 2-5  General Utility Commands Equivalences

<table>
<thead>
<tr>
<th>Utility</th>
<th>Command</th>
<th>Equivalent in Target Environment</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEBCOPY</td>
<td>ALTERMOD</td>
<td></td>
<td>N.R.</td>
</tr>
<tr>
<td></td>
<td>COPY</td>
<td>m_DirCopy</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>COPYGRP</td>
<td>N.R.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COPYMOD</td>
<td>N.R.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXCLUDE</td>
<td>m_DirCopy -e</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>SELECT</td>
<td>m_DirCopy -s</td>
<td>supported</td>
</tr>
<tr>
<td>IEBGENER</td>
<td>With SYSIN DUMMY</td>
<td>m_FileLoad</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>GENERATE MEMBER</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENERATE RECORD</td>
<td>m_FileSort</td>
<td>supported</td>
</tr>
<tr>
<td>IEFBR14</td>
<td>m_ProgramExec IEFBR14</td>
<td>m_ProgramExec IEFBR14</td>
<td>supported</td>
</tr>
<tr>
<td>IEFBR15</td>
<td>m_ProgramExec IEFBR15</td>
<td>m_ProgramExec IEFBR15</td>
<td>supported</td>
</tr>
<tr>
<td>PKZIP</td>
<td>Main parameters</td>
<td>m_Pkzip</td>
<td>supported</td>
</tr>
<tr>
<td>PKUNZIP</td>
<td>Main parameters</td>
<td>m_Pkunzip</td>
<td>supported</td>
</tr>
<tr>
<td>SMTP</td>
<td>m_Ssmtp</td>
<td>m_Ssmtp</td>
<td>supported</td>
</tr>
<tr>
<td>XMITIP</td>
<td>m_SendMail</td>
<td>m_SendMail</td>
<td>supported</td>
</tr>
<tr>
<td>ZIP390</td>
<td>ACTION : ZIP</td>
<td>m_Pkzip</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>ACTION : UNZIP</td>
<td>m_Pkunzip</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>ENCRYPT</td>
<td>N.S.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IFILE</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OFILE</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARCHIVE</td>
<td>supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OVERWRITE</td>
<td>supported</td>
<td></td>
</tr>
</tbody>
</table>
## Sort Utilities

### Sort Utilities Equivalence Tables

**SORT, SORTD, DFSORT, ICEMAN, SYNCSORTT**

Table 2-6  Sort Utilities Equivalences

<table>
<thead>
<tr>
<th>Sort Statement</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>MFSORT</th>
<th>SyncSort</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORT</td>
<td>m_FileSort</td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>MERGE</td>
<td>m_FileSort</td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>OPTION</td>
<td>m_FileSort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>SKIPREC</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>STOPAFT</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>INCLUDE</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>OMIT</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>OUTFILE</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>OUTREC</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>RECORD</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>SUM</td>
<td></td>
<td></td>
<td>Supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>
ICETOOL

Table 2-7 ICETOOL Equivalences

<table>
<thead>
<tr>
<th>Sort Statement</th>
<th>Equivalent in Target Environment</th>
<th>MFSORT</th>
<th>SyncSort</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORT</td>
<td>m_FileSort</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>COPY</td>
<td>m_FileSort</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>MERGE</td>
<td>m_FileSort</td>
<td>supported</td>
<td>supported</td>
</tr>
</tbody>
</table>

Sort Limitations

When using syncsort utility, there are some limitations:

- DFSORT (DATE3) format has no equivalent on syncsort.
- DFSORT statements (outrec, inrec) are not fully supported in the following situations:
  - If a key or a field definition of an outrec field cut a field constant type, an error shows up (MT_ERROR in SYSIN translation) and :#PHX in ksh.
  - If a key or a field definition of an outrec field is not included in the inrec, an error shows up.
  - Cases with SUM are not handled.

DB2 Utilities

Only DB2 Connect 9.7 or above is supported.

Table 2-8 DB2 Utilities

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Oracle</th>
<th>DB2LUW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNTEP2</td>
<td>SQL</td>
<td>m_ExecSQL</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>DSNTEP4</td>
<td>SQL</td>
<td>m_ExecSQL</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>DSNTIAD</td>
<td>SQL</td>
<td>m_ExecSQL</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td>DSNTIAUL</td>
<td>SQL</td>
<td>m_ExecSQL</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>&quot;Unload&quot;</td>
<td>m_DBTableUnload</td>
<td>supported</td>
<td>supported</td>
</tr>
</tbody>
</table>
Table 2-8  DB2 Utilities

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>Equivalent in Target Environment</th>
<th>Oracle</th>
<th>DB2LUW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNUTILB</td>
<td>EXEC SQL</td>
<td>m_ExecSQL</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>m_DBTableLoad</td>
<td>supported</td>
<td>supported</td>
</tr>
<tr>
<td></td>
<td>UNLOAD</td>
<td>m_DBTableUnload</td>
<td>supported</td>
<td>supported</td>
</tr>
</tbody>
</table>
Z/OS JCL in the Batch Runtime Environment
Oracle Tuxedo Application Runtime for Batch Functions

Introduction to the Batch Runtime Commands

This chapter describes:

- The format and rules for using the Batch Runtime Korn shell scripts to run jobs in Emulating z/OS JCL Logic and Architecture.
- The use of the Batch Runtime spawner (EJR) to launch jobs in EJR Syntax.
- The log files and return codes used by the scripts and spawner in Log File Management and Return Code Management.
- The use of the Batch Runtime COBOL runtime to trap errors and manage database interaction in COBOL Runtime.
- A complete description of the Batch Runtime functions in Oracle Tuxedo Application Runtime for Batch Functions.

Emulating z/OS JCL Logic and Architecture

Oracle Tuxedo Application Runtime for Batch provides a set of high-level functions that simplify script syntax enabling more readable and more easily maintainable Korn shell scripts.

Using these functions ensures consistent services; when used together, execution of one function can be conditional on the value of the return code produced by a preceding function.

A function is generally called directly from a Korn shell script resulting from JCL conversion.
Oracle Tuxedo Application Runtime for Batch normalizes Korn shell script formats by proposing a script model where the different execution phases of a job are clearly identified.
EJR Syntax

Synopsis


For TuxJes mode:

EJR (-i|-c|-e|-k) -j JOBID -r JES2ROOT [-l] [-J] Job

Arguments

-c
Conversion phase (only available with TuxJes).

-d regexp
Debug mode — with a regular expression (regexp) describing the functions to debug, for
example -d "m_FileLoad" to debug the m_FileLoad function.

-e
Execution phase (only available with TuxJes).

-f
Environment file (not supported).

-H
Execution stack (use limited to maintenance team).

-i
Input phase (only available with TuxJes).

-j
Job identifier (only available with TuxJes).

-J
Job type - the job type is JCL.

-k
Cancel command (only available with TuxJes).

-l
KSH listing included in log.
-L
Log directory if the environment variable \texttt{MT\_LOG} is defined, else the log is redirected to
TMP directory (without TuxJes).

-r
JES2 root (only available with TuxJes).

-R
Resubmit the job to start a given step.

-s
Sysout directory if the environment variable \texttt{MT\_SYSOUT} is defined, else the sysout files
are redirected to LOG directory and subsequently to TMP directory (without TuxJes).

-t file
Test mode — this option runs the script without executing the different steps. It allows to
check the kinematics of the Korn shell script (use limited to maintenance team).

-v
Verbose mode — the execution report is displayed on screen during execution. By default
not activated.

-V n
Level mode (0 to 9).

Job
Job name — the name of the script to be launched without the \texttt{.ksh} extension.

JOBID
Unique identifier of the job in JES2 system.

JES2ROOT
Root directory of JES2 directory.

**Tools for Managing the Execution of Jobs**

**Log File Management**
When a script is launched with EJR, a log file is generated. When not using TuxJES, the name of
the log file is:

\texttt{JobName\_YYYYMMDDHHMMS\_Jobid.log}. 

The log file is created in a directory identified by the MT_LOG environment variable. The contents of this file provide the production team with detailed information about the execution of a job.

When using TuxJES, refer to the related documentation.

**Return Code Management**

Oracle Tuxedo Application Runtime for Batch uses several return-code variables to manage the result of a function execution and the result of job execution.

<table>
<thead>
<tr>
<th>Table 3-1</th>
<th>Oracle Tuxedo Application Runtime for Batch Return Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Return Code</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>MT_RC</td>
<td>The Return code for an Oracle Tuxedo Application Runtime for Batch function execution.</td>
</tr>
<tr>
<td></td>
<td>• if MT_RC = Cnnnn, return code OK.</td>
</tr>
<tr>
<td></td>
<td>• if MT_RC = Unnnn, User abort</td>
</tr>
<tr>
<td></td>
<td>• if MT_RC= Syyy, System abort</td>
</tr>
<tr>
<td>MT_RC_JOB</td>
<td>General return code (for the job) MT_RC_JOB is updated with MT_RC at the end of each phase. It contains the maximum MT_RC value for the job.</td>
</tr>
<tr>
<td>MT_RC_ABORT</td>
<td>Value fixed to D0000.</td>
</tr>
<tr>
<td></td>
<td>The MT_RC return code is compared to this value to determine if the result of the execution was normal or not.</td>
</tr>
<tr>
<td>MT_RC_PROGRAM_ABORT</td>
<td>By default 127.</td>
</tr>
<tr>
<td></td>
<td>Contains the return code from an executable (not applicative COBOL program).</td>
</tr>
<tr>
<td></td>
<td>• from 0 to 127: return code OK.</td>
</tr>
<tr>
<td></td>
<td>• from 128 to 142: it is a signal SKIL in MT_RC</td>
</tr>
<tr>
<td></td>
<td>• more than 143: Unnn in MT_RC</td>
</tr>
<tr>
<td>MT_RC_STEP_RETURNCODE_{LABEL}</td>
<td>Each phase return code is saved. The variable name contains the phase label. This variable can be used for specific chaining within the script.</td>
</tr>
</tbody>
</table>
### Table 3-2  EJR Return Code

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal end</td>
</tr>
<tr>
<td></td>
<td>Normal execution with user error</td>
</tr>
<tr>
<td>1</td>
<td>System error (abort)</td>
</tr>
<tr>
<td>5</td>
<td>No present of JOBID on command line</td>
</tr>
<tr>
<td>6</td>
<td>No present of JES2ROOT on command line</td>
</tr>
<tr>
<td>11</td>
<td>Some files or directories used for JES2 are not found</td>
</tr>
<tr>
<td>12</td>
<td>Access right problem</td>
</tr>
<tr>
<td>13</td>
<td>Write error - File is not created or written</td>
</tr>
<tr>
<td>21</td>
<td>The KSH script is not found</td>
</tr>
<tr>
<td>22</td>
<td>The KSH script is not readable</td>
</tr>
<tr>
<td>23</td>
<td>The KSH script is not executable</td>
</tr>
<tr>
<td>50</td>
<td>Cancel by user</td>
</tr>
<tr>
<td>51</td>
<td>Cancel by system</td>
</tr>
<tr>
<td>87</td>
<td>PID is not running (killing process)</td>
</tr>
<tr>
<td>52 &lt;= RC &lt;= 99</td>
<td>Temporary Runtime error (the error may appear in case of restart)</td>
</tr>
<tr>
<td></td>
<td>For example: no enough space in technical temporary storage</td>
</tr>
<tr>
<td>123</td>
<td>Default return code must be changed during treatment</td>
</tr>
<tr>
<td>125</td>
<td>Bad verbose level (in case of using option \texttt{V})</td>
</tr>
<tr>
<td>&gt; =100</td>
<td>Permanent Runtime error (the error appears in case of restart)</td>
</tr>
<tr>
<td></td>
<td>For example: an missing of batch RT component</td>
</tr>
</tbody>
</table>
COBOL Runtime

A COBOL runtime, runb, is provided to initialize the execution context of a user COBOL program before the call for its execution. This runtime is used instead of the standard COBOL runtime.

Oracle Tuxedo Application Runtime for Batch Purpose

- Abort trapping procedure definition (standard: std_proc_error and database: dba_proc_error). The standard procedure traps COBOL errors and traces them in a log file. The Database procedure executes a rollback function to insure data integrity.
- Database access function tracing management (mw_dbstat).
- Database Connection and Disconnection and data integrity control (COMMIT and ROLLBACK) if the program is run (m_ProgramExec) with -b option.
- COBOL program execution.

DataBase Interaction Management

Oracle Tuxedo Application Runtime for Batch takes care of the Database context usage:

- Initialization: If a COBOL program is executed (m_ProgramExec) using the -b option, the runtime command, runb, connects it to the database according to the MT_DB_LOGIN environment variable. It must have the correct value (user name, password and Oracle instance, at least "/"). It can be used in the TuxJES Security Configuration file and specified when submitting jobs or set as an environment variable. If the configuration file is not specified, the environment variable MT_DB_LOGIN value is used.

- Termination: Depending on the program return code, the Batch Runtime executes a COMMIT (MT_RC_JOB = 0) or a ROLLBACK (MT_RC_JOB != 0), then disconnects from the database.

Job Restrictions

Oracle Tuxedo Application Runtime for Batch doesn’t support multiple jobs per JCL.

Testing the Validity of a Script (non-exec mode)

Tip: This feature is reserved for the maintenance team.
Using the `-t` argument, it is possible to run the KSH script without executing the internal functions. The `-t` argument allows a script to be checked (for example a newly-developed script) and verify the chaining of the different phases.

Oracle Tuxedo Application Runtime for Batch Functions

Naming Convention

The names of the Batch Runtime functions respect the following format:

```
prefix_ObjectAction
```

Where:

prefix_

- `m` specifies an external function.
- `mi` specifies an internal function.

Object

is the type of object on which the function is used and

Action

is the action to be executed on the object.

Examples include:

- `m_FileAssign`
- `m_FileBuild`
- `m_RcTest`
- `m_ProgramExec`

Reference Page Command Syntax

Unless otherwise noted, commands described in the Synopsis section of a reference page accept options and other arguments according to the following syntax and should be interpreted as explained below.

```
name [ -option . . . ] [cmdarg . . . ]
```

where `name` is the name of an executable file and `option` is a string of one of the following two types: `noargletter`... or `argletter optarg [, . . . ]`
An option is always preceded by a "-".

**noargletter**

A single letter representing an option that requires no option-argument. More than one noargletter can be grouped after a "-".

**optarg**

A character string that satisfies a preceding argletter. Multiple optargs following a single argletter must be separated by commas, or separated by white space and enclosed in quotes.

**cmdarg**

A pathname (or other command argument) that represents an operand of the command.

- (dash) By itself means that additional arguments are provided in the standard input.

-- (two dashes) Means that what follows are arguments for a subordinate program.

[ ]

Surrounding an option or cmdarg, mean that the option or argument is not required.

{}  

Surrounding cmdargs that are separated by an or sign, mean that one of the choices must be selected if the associated option is used.

"OR" argument

. . .

Means that multiple occurrences of the option or cmdarg are permitted.
Reference

The Oracle Tuxedo Application Runtime for Batch Reference Guide describes, in alphabetic order, shell-level functions delivered with the Batch Runtime software.

The following functions are described:

Table 3-3  Batch Runtime Functions

<table>
<thead>
<tr>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_CondElse</td>
</tr>
<tr>
<td>m_CondEndif</td>
</tr>
<tr>
<td>m_CondExec</td>
</tr>
<tr>
<td>m_CondIf</td>
</tr>
<tr>
<td>m_DBTableLoad</td>
</tr>
<tr>
<td>m_DirCreate</td>
</tr>
<tr>
<td>m_DirDelete</td>
</tr>
<tr>
<td>m_DirRename</td>
</tr>
<tr>
<td>m_ExecSQL</td>
</tr>
<tr>
<td>m_ExpdtAlter</td>
</tr>
<tr>
<td>m_FileAssign</td>
</tr>
<tr>
<td>m_FileBuild</td>
</tr>
<tr>
<td>m_FileClrData</td>
</tr>
<tr>
<td>m_FileDelete</td>
</tr>
<tr>
<td>m_FileEmpty</td>
</tr>
</tbody>
</table>
Overview

The functions correspond to the interface (API) between the shell script and the Batch Runtime executable. Some scripts, such as m_JcllibSet, are used only in the conversion stage and are not present in the extended script that is available for execution.
**m_CondElse**

**Name**
m_CondElse — Else of a condition.

**Synopsis**
m_CondElse

**Description**
This function marks the alternative part of a m_CondIf function.

**Options**
No parameters.
m_CondEndif

Name
m_CondEndif - End of a condition

Synopsis
m_CondEndif

Description
This function ends the previous IF condition.

Options
No parameters.
m_CondExec

Name
m_CondExec — conditional execution (for a program or procedure).

Synopsis
m_CondExec condexp [condexp...]

Description
Conditional execution. If condition is true, the remaining command in the current step is ignored. Each condition expression contains either, EVEN, ONLY or a value, operator[,step] condition. An m_CondExec statement may contain several condition expressions and this association specifies a logical "OR" of the different conditions.

Options
Condexp [condexp]
Condition expression.

   EVEN
     Executes step even if previous step ended abnormally.

   ONLY
     Executes step only if previous step ended abnormally.

value, operator[,step]
Where <step> is any of the previous steps. If the previous step was not executed, the condition is false. When <step> refers to a previous step, replace it with the step label, e.g. "STEPEC01". When <step> refers to a return code of the step in the procedure, replace it with "STEP_PROCNAME_NUM", where "PROCNAME" indicates the name of procedure, and "NUM" indicates the sequence number of the procedure calls.

Examples
m_CondExec EVEN

To refer to a step:

m_CondExec 4,LT,STEPEC01 8,LT,STEPEC02 ONLY

To refer to a return code of the step in procedure:
m_CondExec 4, LT, STEP_PROCNAME_NUM ONLY

Where PROCNAME is the name of procedure, and NUM is the sequence number of procedure calls.

m_CondIf

Name
m_CondIf - Conditional execution

Synopsis
m_CondIf "condexp [condexp...]"

Description
Executes the condition contained in the "condexp" parameter. Nested levels of "if" conditions are authorized.

Options
"Condexp [condexp]"
   Conditional expression.
   RC,operator,value
      RC indicates a return code.
   STEP.RC,operator,value
      STEP.RC indicates that the expression tests a return code for a specific STEP.
      Operator indicates the operator used for the conditional expression (GT, LT, EQ etc.).
   STEP.RUN,EQ/NE,TURE/FALSE
      STEP.RUN indicates that a specific job step started execution or not.
   ABEND
      ABEND indicates an abend condition occurred.
   ABENDCC=number
      ABENDCC indicates a system or user completion code.

Examples
m_CondIf " RC, EQ, 3"
**Note:** The statements following this `m_CondIf` statement are executed if the return code is equal to 3.
**m_DBTableLoad**

**Name**
m_DBTableLoad – Loads the content of an input file into a database table.

**Synopsis**
m_DBTableLoad -t -i [-e] [-d] [-D]

**Description**
This function executes a command stored in the file which ddname is MT_CTL.
This command is either an SQLLDR command for Oracle or a DB2 LOAD command for DB2LUW according to the target database.

**Options**
- `-t <table name>`
  Mandatory option.
  The name of the database table to be loaded.

- `-i <input flat file>`
  Mandatory option.
  The ddname of the file containing the data to be loaded.

**Options for Oracle**
- `-e <error file>`
  Optional.
  SYSERR by default.
  The ddname of the error file where are stored the errors during sqlldr command execution.

- `-d <discard file>`
  Optional.
  SYSERR by default.
  The ddname of the file that contains the discarded data during sqlldr command execution.

- `-D <discard file>`
  Optional.
  999 by default
The maximum number of discarded data.

Options for DB2LUW

-e <error file>
Optional.

<Logfile> by default.
The ddname of the log file used during db2 load command execution.

-d <discard file>
Ignored.

-D <discard file>
Ignored.

Examples

```
m_FileAssign -d OLD DDIN ${DATA}/MYINPUTFILE
m_FileAssign -d SHR MT_CTL ${MT_CTL_FILES}/MYLOADCTL
m_FileAssign -d SHR SYSERR ${[DATA}/MYSYSERR
m_DBTableLoad -t MYTABLE -i DDIN
```
m_DBTableUnload

Name
m_DBTableUnload – Unloads the content of a database table into a flat file.

Synopsis
m_DBTableUnload -t -o

Description
This function executes a command stored in the file which ddname is MT_CTL.
This command is either an SQLPLUS command for Oracle or a DB2 EXPORT command for
DB2LUW according to the target database.

Options
-<table name>
  Mandatory option.
  The name of the database table to be unloaded.
-o <output flat file>
  Mandatory option.
  The ddname of the file containing the unloaded data.

Examples

  m_FileAssign -d OLD DDOUT $(DATA)/MYOUTPUTFILE
  m_FileAssign -d SHR MT_CTL $(MT_CTL_FILES)/MYUNLOADCTL
  m_DBTableUnload -t MYTABLE -o DDOUT
m_DirCopy

Name
m_DirCopy – Copies the members of a directory.

Synopsis
m_DirCopy [-i] [-o] [-s] [-e]

Description
This function copies the members of a directory to another directory.

Options
- i [(] { input | (input,R) } [:...][)]
  Optional.
  Default value: SYSUT1.
  input: ddname of the input directory to be copied.
  R: specifies that all members to be copied are to replace any identically named members in the output directory. When this option is specified, the ddname and R parameter must be enclosed in a set of parentheses.
  Several directories may appear separated by “:”

- o <output dir>
  Optional.
  Default value: SYSUT2.
  The ddname of the output directory.

- s <member list>
  Optional.
  Member list to be copied (separated by comma and enclosed in a set of parentheses).

- e <member list>
  Optional.
  Member list to be excluded (separated by comma and enclosed in a set of parentheses).
Examples

Copy of all the members from directory PDS1 to directory SEQ1.

```bash
m_DirCopy -i PDS1 -o SEQ1
```

Copy of the members A and K from directory PDS1 to directory SEQ1.

```bash
m_DirCopy -i PDS1 -s "(A,K)" -o SEQ1
```

Copy of all the members except member A from directory PDS1 to directory SEQ1.

```bash
m_DirCopy -i PDS1 -e A -o SEQ1
```
**m_DirCreate**

**Name**
m_DirCreate – Creates a directory.

**Synopsis**
m_DirCreate [-v volume] DirName

**Description**
This function creates a directory.

**Options**

- **-v volume**
  Indicates the directory volume. This option only works when file catalog is enabled in Batch Runtime.

- **DirName**
  The name of the directory to be created.
**m_DirDelete**

**Name**
m_DirDelete – Deletes a directory.

**Synopsis**
m_DirDelete [-v volume] DirName

**Description**
This function deletes a directory.

**Options**

- **-v volume**
  Indicates the directory volume. This option only works when file catalog is enabled in Batch Runtime.

**DirName**
The name of the directory to be deleted.
**m_DirRename**

**Name**
m_DirRename – Renames a directory.

**Synopsis**
m_DirRename [-v volume] OldDirName NewDirName

**Description**
This function renames a directory.

**Options**

- **-v volume**
  Indicates the new directory volume. This option only works when file catalog is enabled in Batch Runtime.

- **OldDirName**
  The name of the directory to be renamed.

- **NewDirName**
  The new name of the directory to be renamed.
m_ExecSQL

Name
m_ExecSQL — Executes an SQL script.

Synopsis
m_ExecSQL [-b RC0] [-f] [-o]

Description
This function executes an SQL script.
The SQL directives (CREATE TABLE, CREATE INDEX, DELETE, SELECT ...) are in the
<inputfile> file. The results will be stored in the file <outputfile>.

Options

-b RC0
With -b RC0, whether encountering errors or not, all SQL will be executed and then all
successful SQL will be committed in the end; however, without -b RC0, if any error is
encountered, SQL will rollback when terminating.

-f <inputfile>
The <inputfile> will contain the SQL directives (ddname of the file).
SYSIN is the default value.

-o <outputfile>
The <outputfile> will contain the results (ddname of the file).
SYSPRINT is the default value.

Examples

In the first example, the SQL directives are in the in-stream SYSIN file, the results will be stored
in the SYSREC00 file.

m_FileAssign -d ,CATLG SYSREC00 $(DATA)/FBACKE.LST.CUMUL
m_FileAssign -i SYSIN
SELECT * FROM PJ01DB2.TABTEST2;
m_ExecSQL -o SYSREC00

In the second example, the SQL directives are the file TOW132C.sysin and the results will be printed.

m_OutputAssign -c "*" SYSPRINT
m_FileAssign -d SHR SYSIN $(SYSIN)/SYSIN/TOW132C.sysin
m_ExecSQL

**Note:** The DB2 commands are not translated. The user has to verify these commands according to the target database software.

### m_ExpdtAlter

**Name**

m_ExpdtAlter -- modify the expiration date of a data set.

**Synopsis**

```
m_ExpdtAlter [-m Ydate|Ddays] dsname
```

**Description**

This function modifies the expiration date the data set.

**Options**

- **-m <Ydate|Ddays> (Y upper case character) (D upper case character)**
  
  Specifies the retention period for the data set.

- **date**
  
  Specifies the expiration date for the data set. The date is specified in the form [yy]yyddd.

  yyyddd
  
  The yyyy is a four-digit year (through 2155), and ddd is the three-digit day of the year.

yyddd
The yy is a two-digit year number (through 99) and the ddd is a three-digit day number from 000 through 365 for non-leap year dates. For example, code February 2, 1995 as 95033

days
Specifies the retention period for the data set. It is specified in the form dddd, where dddd should be set in the range of 0-9999. The system adds dddd to the current date to produce an expiration date.

**m_FileAssign**

Name
m_FileAssign — Assigns a file.

Synopsis
m_FileAssign -d DISPMODE [-g [CUR|ALL [+-np][rang]]] [-C] [-r RecSize -t Type [-k Key]]|S ModeFile| [-v volume] [-e Ydate|Ddays] [-w Reference] DDNAME DSNAME

Description
m_FileAssign assigns a file. If assigning a file triggers the creation of a file, the creation process precedes the assign itself.

Specific cases are:

- New files (*DISP=NEW* parameter).
- Concatenated files (DD cards, where only the first one contains a label). In this case a concatenation is made in a temporary file, the original DSNAME is replaced by the name of the temporary file.
- Override files (file override in the JCL); a specific assign function m_FileOverride is used. This function call is implanted in each STEP required, before the execution of the program.
- In the case where a file assign contains a *DISP=NEW, DELETE, DELETE* parameter, a delete process is added to the end (normal and abnormal) of the step.
For the DISP=OLD and DISP=PASS options, the file is kept.

For the DISP=MOD option, the write to the file is made in a temporary intermediary file, then by a copy in Extend on the original file.

In one step, if a DDNAME is defined more than one time by m_FileAssign, only the first one takes effect. The disposition specified for the former DSNAME is also discarded.

**Options**

At least one of the two options “-i” and “-d” must be specified. All other options are optional.

- **-C <concatenation>**
  Concatenate this file with the previous dsname for this ddname.

- **-D <delimiter>**
  Delimiter of sysin.

- **-d <DispOption>**
  This option indicates the DISPosition status of the file in the format:
  
  DISP=[(status)[,normal-termination-disp][,abnormal-termination-disp]]
  
  Possible combinations are:

  ```
  DISP= ( [NEW] [,DELETE ] [,DELETE ] )
  [OLD] [,KEEP] [,KEEP ]
  [SHR] [,PASS] [,CATLG ]
  [MOD] [,CATLG ] [,UNCATLG ]
  [ANY] [,UNCATLG ]
  ```

  The Disp Option indicates the status of the data set at the beginning of a job step and what to do with the data set in the event of normal and abnormal termination of the step.

  **<status>**
  
  The status indicates if an existing data set should be used or a new one created. For existing data sets the status indicates if the data set can be shared with other jobs or used to append records to the end of the data set. The possible values are:
  NEW — indicates to create a new unshared data set.
OLD — indicates to use an existing unshared data set.
SHR — indicates to use an existing shared data set.
MOD — indicates an existing unshared data set to add records at the end of file.

An additional status has been added to the traditional z/OS status:
ANY — indicates to use a file in a special mode. The other sub-parameters (<normal-termination-disp> and <abnormal-termination-disp>) are ignored in this case.

Note: OLD and SHR check the file is already existing.
Note: NEW, creation if the file does not exist, abort if the file already exists.
MOD, creation if the file does not exist, continue if the file already exists.

<normal-termination-disp>
This option indicates what to do with a data set when a step ends normally. The possible values are:
DELETE — The data set is no longer needed.
KEEP — The data set is to be kept.
PASS — The data set is to be passed for use by a subsequent step.
CATLG — The data set is to be cataloged.
UNCATLG — The data set is to be deleted from catalog.

<abnormal-termination-disp>
DELETE — The data set is no longer needed.
KEEP — The data set is to be kept.
CATLG — The data set is to be cataloged.
UNCATLG — The data set is to be deleted from catalog.

The termination dispositions have default values for each status, the default values are:
NEW: DELETE, DELETE
OLD/SHR/MOD: KEEP, KEEP

Note: In one step, PASS is functionally equivalent to KEEP; however, in one job, if a data set is newly created by NEW or MOD, and no KEEP or CATLG in the following steps is specified to this data set, this data set will be deleted.

Note: Once specified for a GDS, KEEP becomes a member of GDG after job execution.

Note: For GDG, CATLG and UNCATLG are converted to KEEP implicitly, and KEEP always results in the newly created GDS becoming a member of GDG.

-g <generationFile>
Indicates that the data set is a generation file. The possible values are:
+n: creates the nth new generation file.
-n: accesses the nth previous old generation file.

0: accesses the current generation.

cur: accesses the current generation.

all: concatenates all generations.

**Note:** If "all" is specified, all generation files for the corresponding GDG, including all
new added GDS, are concatenated to one temporary file, and this temporary file
is assigned to DD specified for m_FileAssign. For example, if "all" is specified
and <normal-termination-disp> or <abnormal-termination-disp> is specified to
DELETE, all generation files are deleted from the corresponding GDG in the end
of the current step.

-i

Indicates that the data set is a sysin.

-k <key position=>key length>

Indicates the key characteristics of an indexed file (to be used when file organization is
IDX).

-r [<record length-min>]<record length-max>

Indicates the length characteristics of the record.

For fixed files, only the maximum record length is specified.

For variable files, the minimum and the maximum record lengths are specified.

-S <model file>

(S upper case character)

Names the dsname of a file.

When the disposition mode is NEW, this file is a “model” for the new file to be created.

The characteristics of the new file depend on the characteristics of the “model” file and
other parameter given by the m_FileAssign function. IDX, LSEQ, SEQ, REL, and GDG
are supported as model file.

-t <file organization>

Indicates the file organization type:

SEQ: sequential

LSEQ: line sequential
REL: relative
IDX: indexed
PDS: directory

-v volume
Indicates the external file volume. This option only works when file catalog is enabled in Batch Runtime.

-e Ydate|Ddays (Y upper case character) (D upper case character)
Indicates the retention period for the data set.

date
Specifies the expiration date for the data set. The date is specified in the form [yy]yyddd.

yyddd
The yyyy is a four-digit year (through 2155), and ddd is the three-digit day of the year.

yyddd
The yy is a two-digit year number (through 99) and the ddd is a three-digit day number from 000 through 365 for non-leap year dates. For example, code February 2, 1995 as 95033

days
Specifies the retention period for the data set. It is specified in the form dddd, where dddd should be set in the range of 0-9999. The system adds dddd to the current date to produce an expiration date.

-w referenced-file
Indicates that the volume of the referenced file is used. This option only works when file catalog is enabled in Batch Runtime.

The reference mode can be one of the following values.
- FileName
- *.DD
- *.STEP.DD
**ddname <InternalFileName>**
The logical name of the file as defined in the SELECT statement of the COBOL program.

**dsname <ExternalFileName>**
Real file name, full path of the file on the disk.

**Examples**

Example with a shared file:

```bash
m_FileAssign -d SHR ENTREE $(DATA)/PJ01DDD.BT.QSAM.KBIEI001
```

Example with a sysin and a delimiter:

```bash
m_FileAssign -i -D FF INPUT
data input 1
data input 2
FF
```

Example with a sysin and continuation:

```bash
m_FileAssign -i SYSIN
data input 1
_end
#%OPC BEGIN ACTION=INCLUDE
   m_FileAssign -i -C
data input 2
_end
#%OPC END ACTION=INCLUDE
   m_FileAssign -i -C
data input 3
_end
```

Example with a new file:

```bash
m_FileAssign -d NEW -r 188 -t SEQ ENTREE $(DATA)/PJ01DDD.BT.KBIEI001
```
The new file will be a sequential file with a fixed record length of 188 bytes.

Example with a "model" file:

```
m_FileAssign -d NEW -r 188 -S PJ01.MODEL ENTREE $(DATA)/PJ01.OUTPUT
```

The new file have the characteristics of the file PJ01.MODEL except for the record length given by the "-r" parameter.

---

**m_FileBuild**

**Name**

`m_FileBuild` — Creates a file.

**Synopsis**

```
```

**Description**

This function creates a file.

**Options**

- `-t <organization>`
  
  Type is the organization type of the created file. The possible values are:

  - SEQ
    
    for a sequential file.

  - LSEQ
    
    for a line sequential file.

  - IDX
    
    for an indexed file

  - REL
    
    for a relative file

  **Note:** The options `-r` and `-k` are mandatory for an indexed file.
-r Length
   Indicates the record length of the file. This option is mandatory for indexed files.

-k Position+Length
   The primary key (mandatory for indexed files)
   Position
      The first character of the key in relation to the beginning of the record.
   Length
      The length of the primary key.

-K Position+Length[d]
   (K upper case character)
   The secondary key indicating that the file contains a secondary key.
   Position
      The first character of the key in relation to the beginning of the record.
   Length
      The length of the secondary key.
   d
      Optional parameter.
      Permit duplicates of secondary key.

-S <model file>
   (S upper case character)
   Names the dsname of a file.
   When the disposition mode is NEW, this file is a “model” for the new file to be created.
   The characteristics of the new file depends on the characteristics of the “model” file and
   other parameter given by the m_FileAssign function.

Note: Model files of type "GDG|RDB|PDS" are NOT supported.

-v volume
   Indicates the file volume. The file is cataloged automatically. This option only works
   when file catalog is enabled in Batch Runtime.

-e Ydate|Ddays (Y upper case character) (D upper case character)
   Indicates the retention period for the data set.
**date**

Specifies the expiration date for the data set. The date is specified in the form [yy]yyddd.

**yyddd**

The yy is a two-digit year number (through 99) and the ddd is a three-digit day number from 000 through 365 for non-leap year dates. For example, code February 2, 1995 as 95033

**yyyddd**

The yyyy is a four-digit year (through 2155), and ddd is the three-digit day of the year.


**days**

Specifies the retention period for the data set. It is specified in the form dddd, where dddd should be set in the range of 0-9999. The system adds dddd to the current date to produce an expiration date.

**dsname**

dsname of the file to create.

**Examples**

To build an indexed file with no secondary key, the following function builds an indexed file with a record length of 266 bytes. There is no secondary key and the primary key begins in the first character of the record and is six characters long:

```
m_FileBuild -t IDX -r 266 -k 1+6 ${DATA}/METAW00.VSAM.CUSTOMER
```

To build a similar indexed file, with in addition, a non-duplicate secondary key in position 20 with a length of 7 the following function can be used:

```
m_FileBuild -t IDX -r 266 -k 1+6 -K 20+7 ${DATA}/META00.VSAM.CUSTOMER
```

To build a similar indexed file with a secondary key allowing duplicates in position 20 with a length of 7 the following function can be used:

```
m_FileBuild -t IDX -r 266 -k 1+6 -K 20+7d ${DATA}/META0.VSAM.CUSTOMER
```
m_FileClrData

Name

m_FileClrData - clears a file.

Synopsis

m_FileClrData [-v volume] FileName

Description

m_FileClrData is used to clear a file. GDG file is not supported.

Options

-v volume  
Indicates the file volume. This option only works when file catalog is enabled in Batch Runtime.

FileName  
The name of the file to be cleared.

Example

m_FileClrData ${DATA}/PJ01DDD.BT.QSAM.KBST0045
**m_FileDelete**

**Name**

*m_FileDelete* — Deletes a file.

**Synopsis**

```
m_FileDelete [-v volume] FileName
```

**Description**

*m_FileDelete* is used to delete a file.

**Options**

- `-v volume`

  Indicates the file volume. The file is uncataloged after deletion. This option only works when file catalog is enabled in Batch Runtime.

**FileName**

The name of the file to be deleted.

**Example**

```
m_FileDelete ${DATA}/PJ01DDD.BT.QSAM.KBST0045
```
m_FileEmpty

Name
m_FileEmpty – Checks whether a file is empty.

Synopsis
m_FileEmpty -r ReturnVariable [-v volume] FileName

Description
m_FileEmpty is used to check whether a file is empty.

Options
- r ReturnVariable
  Returns “true” or “false”.

- v volume
  Indicates the file volume. This option only works when file catalog is enabled in Batch Runtime.

FileName
The name of the file to be checked.

Example
m_FileEmpty -r MY_VARIABLE $(DATA)/rextest2
if [[ $(MY_VARIABLE) = true ]]; then
echo "file is empty"
else
  echo "file is not empty"
fi
m_FileExist

Name
m_FileExist – Checks the presence of a file.

Synopsis
m_FileExist -r ReturnVariable [-v volume] FileName

Description
m_FileExist is used to check whether a file is present.

Options

-r ReturnVariable
  Returns “true” or “false”.

-v volume
  Indicates the file volume. This option only works when file catalog is enabled in Batch Runtime.

FileName
  The name of the file to be checked.

Example
m_fileExist -r MY_VARIABLE $(DATA)/rextest2
if [[ $(MY_VARIABLE) = true ]]; then
  echo "file exists"
else
  echo "file does not exist"
fi

m_FileListcat

Name
m_FileListcat -- query the catalog according input.
Synopsis

```
m_FileListcat -e 'entry [entry...]' -l 'level' -x days -o ddname
NAME|HISTORY|VOLUME|ALLOCATION|ALL
```

Description

This function queries the catalog according input, gives outputs.

Options

- **-e**
  
  Specifies the name or generic name of each entry to be listed, space is used as delimiter.

- **-l**
  
  Specifies that all entries that match the level of qualification specified by (level) are to be listed irrespective of the number of additional qualifiers.

**Note:** Parameter -e and -l are exclusive

Examples of -e and -l specifications, suppose a catalog contains the following names:

1. A.A.B
2. A.B.B
3. A.B.B.C
4. A.B.B.C.C
5. A.C.C
6. A.D
7. A.E
8. A

If –e A.* is specified, entries 6 and 7 are listed;
If –e A.*.B is specified, entries 1 and 2 are listed;
If –l A.*.B is specified, entries 3, and 4 are listed;
If –l A is specified, entries 1, 2, 3, 4, 5, 6, and 7 are listed.

- **-x**
  
  Specifies the entries that will expire in the specified number of days or earlier. The maximum number that can be specified is 9999 and it indicates that all entries are to be listed. Zero indicates that only entries that have already expired are to be listed.
-o
  Specifies a data set, other than the SYSPRINT data set, to receive the output produced by LISTCAT.

NAME
  Specifies that the name and entry type of the entries are to be listed.

HISTORY
  Specifies that only the following information is to be listed for each entry: name, entrytype, creation date, expiration date. Abbreviation: HIST.

VOLUME
  Specifies that the information provided by specifying HISTORY, plus the volume, are to be listed. Abbreviation: VOL

ALLOCATION
  Specifies that the information provided by specifying HISTORY, plus the volume, are to be listed. Abbreviation: ALLOC.

ALL
  Specifies that the information provided by specifying HISTORY, plus the volume, are to be listed.

Examples
  m_FileListcat -l 'A.*.B' -x 100 NAME
**m_FileLoad**

**Name**

m_FileLoad — Loads a file.

**Synopsis**

```
m_FileLoad [-C] [-S] [-v volume] Infile [Infile ...] Outfile
```

**Description**

This function loads a file.

**Options**

- **-C**
  
  Number of records to copy from the Infile to the Outfile.

- **-S**
  
  Number of records to skip when copying from the Infile to the Outfile.

- **-v volume**
  
  Indicates the Outfile volume. This option only works when file catalog is enabled in Batch Runtime.

**Example**

```
m_FileLoad ${DD_SYSUT1} ${DD_SYSUT2}
```
m_FileOverride

Name
m_FileOverride — Overrides a file.

Synopsis
m_FileOverride [-i][-d][-g][-r][-k][-S][-t] -s label ddname dsname
m_FileOverride -i [-D Delimiter] -s Labelproc DDNAME
m_FileOverride -d [[-r RecSize -t Type [-k key]]|-S ModelFile][-g [CUR|ALL
[(-|+)np][rang]]] -s label DDNAME DSNAME

Description
m_FileOverride overrides a file assignment, this assign has priority over a standard assign (m_FileAssign).

Options
Only “-s” and one of the two options “-i” and “-d” are mandatory to specify the usage. All other options are optional. See m_FileAssign for other options.

-s <label>
   Name of the label in the called procedure.

Example
m_FileOverride -i -s PR3STEP1 SYSIN

m_FileOverride -d OLD -s MYSORT CUSTOM
${DATA}/BEAUSR2.QSAM.CUSTOM
m_FilePrint

Name
m_FilePrint — Prints a file (IDCAMS command PRINT).

Synopsis
m_FilePrint [-v volume] {-i ddname|-I dsname} {-o ddname|-O dsname}[-t][-C][-S]

Description
This function prints a file.

Options
-C
Optional: Number of records to be listed.

-S
Optional: Number of records to skip before the listing begins.

-i infile|-I indataset
The input file is either a ddname (infile) or a dsname (indataset).

-o ddname|-O dsname
Optional: The output file is either a ddname (outfile) or a dsname (outdataset) (default value: SYSPRINT)

-t {CHAR|DUMP|HEX}
Optional: Type of print (default value: DUMP).

CHAR: specifies each byte of a record is to be listed as character.

DUMP: specifies each byte of a record is to be printed in both hexadecimal and character format.

HEX: specifies each byte of a record is to be listed as hexadecimal digits.

-v volume
Indicates the output file volume. This option only works when file catalog is enabled in Batch Runtime.
Example

m_FilePrint -I $(DATA)/INPUT -C 1
m_FilePrint -i INPUT -t CHAR -C 5
**m_FileRepro**

**Name**
m_FileRepro — Copies a file (IDCAMS command REPRO).

**Synopsis**
m_FileRepro [-v volume] {-i ddname|-I dsname} {-o ddname|-O dsname}[-C] [-S]

**Description**
This function copies a file.

**Options**
- **-C**
  Optional: Number of records to be copied.

- **-S**
  Optional: Number of records to skip before the copy begins.

- **-i infile|-I indataset**
  The input file is either a ddname (infile) or a dsname (indataset).

- **-o ddname|-O dsname**
  Optional: The output file is either a ddname (outfile) or a dsname (outdataset) (default value: SYSPRINT).

- **-v volume**
  Indicates the output file volume. This option only works when file catalog is enabled in Batch Runtime.

**Example**

```
  m_FileRepro -I ${DATA}/INPUT -C 1
  m_FileRepro -i INPUT -C 5
```
**m_FileSort**

**Name**
m_FileSort — Sorts a file.

**Synopsis**
m_FileSort -s SortSpecificationFile -i Infile1[,Infile2 ...] -o [Outfile]

**Description**
This function sorts a file.

**Options**

- **-s SortSpecification File**
  The sort specification indicates either a file containing the sort specification or a file that indicates where the sort specification is to be found (ddname of file, by default SYSIN).

- **-i Infile**
  At least one file must be used as input to the sort (ddname of file, by default SORTIN).

- **-o Outfile**
  File to be used as output to the sort (ddname of file, by default SORTOUT).

**Example with SyncSort commands**

```bash
m_FileAssign -i TOOLIN
/FIELDS FLD1 5 CH 5
/COND ...
/OMIT ...
_end

m_FileSort -s TOOLIN -i SORTIN -o SORTOUT
```
m_FileRename

Name
m_FileRename – Renames a file.

Synopsis
m_FileRename [-v volume] OldName NewName

Description
m_FileRename is used to rename a file.

Note: Renaming for VSAM RDB files is not supported; the result is unpredictable for renaming an RDB file. Renaming for GDG is supported.

Options
-v volume
Indicates the new file volume. This option only works when file catalog is enabled in Batch Runtime.

NewName
The new name of the file.

OldName
The old name of the file.
**m_Ftp**

**Name**
m_Ftp — Executes an FTP process.

**Synopsis**
m_Ftp -i <inputfile> [-e <ExitReturnCode>] [-n NETRC]

**Description**
This function emulates the z/OS feature :EXEC PGM=FTP.

This function launches an ftp process and executes ftp command(s) stored in an input file.

An environment variable MT_FTP_TEST must be declared and initialized with the following value (upper or lower case):

Y or YES : test mode. The ftp commands different from “open”, “user”, “quit” or “bye” will not be executed.

N or NO: real mode. All the ftp commands will be executed.

**Options**

- **-i <inputfile>**
  Mandatory parameter.
  ddname of the file which contains the ftp commands.

- **-e <ExitReturnCode>**
  Optional parameter.
  Numeric return code to be returned when an error occurred during ftp execution.
  If ExitReturnCode = nn, the return code will be “C00nn” in case of ftp error.
  If the parameter does not exist, the return code is always C0000.

- **-n NETRC**
  Optional parameter.
  Necessary when the ftp connection is executed through “.netrc” file.
So “user” and “password” are omitted in <inputfile>.

Examples

```
m_FileAssign -d ANY MT_LOC01 $(DATA)/ftp_file_loc1
m_FileAssign -i SYSIN
open host
user user1 pw1
put MT_LOC01 DIR/file2
quit
_end
```

```
m_Ftp -i INPUT
```

Notes

Some notes concerning the lines in the “SYSIN” file.
- The first line must be “open”:
  format: open <HOSTNAME or ADDRESS>
  example: open 172.20.12.21
- The second line must set the user ID and the password if the “-n NETRC” parameter is not set.
  format: user <user> <password>
  When the “-n NETRC” parameter is present, it implies that a “.netrc” file exist. This file identifies
  the user ID and the password to be used.
- The following lines are ftp’s commands (one per line) as get, put, ...
- The last line must be “quit”:
  format: quit
m_GenDefine

Name
m_GenDefine — Defines a GDG.

Synopsis
m_GenDefine [-v volume][-e Ydate|Ddays] -s [-r] --nb_occurs <GDG base name>

Description
This function defines the GDG file. The execution result depends on the GDG management way defined in MT_GENERATION variable.

Options
- **v volume**
  Indicates the GDG file volume. The GDG file is automatically cataloged. This option only works when file catalog is enabled in Batch Runtime.

- **e Ydate|Ddays (Y upper case character) (D upper case character)**
  Indicates the retention period for the data set.

**date**
Specifies the expiration date for the data set. The date is specified in the form [yy]yyddd.

  - **yyddd**
    The yy is a two-digit year number (through 99) and the ddd is a three-digit day number from 000 through 365 for non-leap year dates. For example, code February 2, 1995 as 95033.

  - **yyyyddd**
    The yyyy is a four-digit year (through 2155), and ddd is the three-digit day of the year.


**days**
Specifies the retention period for the data set. It is specified in the form dddd, where dddd should be set in the range of 0-9999. The system adds dddd to the current date to produce an expiration date.
-s
Number of occurrences of generation file to keep on disk. Its value should be set in the range of 1-9999.

-r
Indicates whether to allow re-defining of a GDG.

If -r is specified and the GDG to be defined already exists, only the maximum number of generation files of the GDG is updated with the value specified by -s.

GDG base name
The name of the GDG base for which the maximum number of generations is being defined. The maximum length of GDG base name is 1024.

Example
m_GenDefine -s 31 $(DATA)/PJ01DDD.BT.GDG.KBIDU001
**m_JclLibSet**

**Name**
m_JclLibSet — Specify conversion stage Procedure and Include directories.

**Synopsis**
m_JclLibSet directory

**Description**
m_JclLibSet specifies the directories where Procedures and Includes are stored during the conversion phase.

**Options**

**directory**
Path and name of the directory.

**Example**

m_JclLibSet PJ01DDD.BT.INCLUDE.SRC
**m_JobBegin**

**Name**
m_JobBegin — Used to begin a job.

**Synopsis**
m_JobBegin -j jobname [-C cond] [-c class] [-p priority] [-r restart] [-t typrun] -v version -s start_label

**Description**
Indicates the parameters that are used on the z/OS job card with the JES2 interface. The parameters are stored in the following files:

- `class` is stored in the `JOBID.class` file
- `restart` is stored in the `JOBID.restart` file
- `priority` is stored in the `JOBID.priority` file
- `typrun` is stored in the `JOBID.typrun` file

**Options**

- `-j jobname`
  The name of the job to launch.

- `-C condition`
  Specifies the return code tests used to determine whether a job will continue processing or be terminated.

- `-c class`
  The execution class of the job.

- `-p priority`
  The execution priority of the job.

- `-r restart`
  The name of the step to use to restart the job.

- `-t typrun`
  Indicates what should be done with the job. One of the following choices:
  
  - `COPY` – Copy the job directly in an output stream to sysout.
HOLD – The system should hold the job.

JCLHOLD – JES2 should hold the job.

SCAN – Scan JCL for syntax errors only.

-v version
version of the ksh script.

-s start_label
Start label — label of the first phase to be started.

Example

m_JobBegin -j PJ01DSTA -s START -v 1.0 -t SCAN
**m_JobEnd**

**Name**

m_JobEnd — Ends a job.

**Synopsis**

m_JobEnd

**Description**

This function is used to end a job.

**Options**

None
m_JobLibSet

Name
m_JobLibSet — Specifies where programs are stored.

Synopsis
m_JobLibSet directory [:directory[:directory...]]

Description
This function specifies at job level the directory in which programs are stored.

Options
directory [:directory[:directory...]]
Path and name of the directory containing executable programs.
**m_OutputAssign**

**Name**

*m_OutputAssign* – manages DD SYSOUT statements with the following parameters: CLASS, COPIES, DEST, FORMS and HOLD.

**Note:** If users want to redirect the output intended for standard output by "DISPLAY" statement in COBOL programs into the file specified by SYSOUT DD:

For MicroFocus COBOL, OUTDD directive must be added while compiling the program; for COBOL-IT COBOL, it's not applicable to do it.

**Synopsis**

*m_OutputAssign* [-c class][-w writer][-n copies][-d dest][-f forms][-H][-o list of output][-D dsname] DDNAME

**Options**

- **-c <class>**
  
  Class of the output queue.

- **-w <writer>**
  
  - INTRDR:
    
    At the end of the *m_ProgramExec* function, submits the contents to TuxJes (hopefully a script shell).

  - SMTP:
    
    At the end of the *m_ProgramExec* function, submits the contents (hopefully SMTP commands) to send an email using the SMTP protocol.

    - <writer>:
      
      At the end of the *m_ProgramExec* function, submits the command associated with the <writer> name in the writer.conf file.

- **-n <copies>**
  
  Number of copies to print.

- **-d <dest>**
  
  Destination of the printing.
-f <forms>
    Name of the used form

-H <Y/N>
    Specifies whether the print must held or not.
    N is the default value.

-o <reference[,reference,...]>
    List of "OUTPUT" references.

-D <dsname>
    Data set name.

**ddname**
    Data Definition Name

**Examples**

Example with an output class A:
    m_OutputAssign -c A SYSOUT

Example with INTRDR:
    In this case, the file which ddnmae is RDRCICO must contain a ksh script.
    m_OutputAssign -c R -w INTRDR RDRCICO

Example with an "output" reference:
    The output EDI is referenced.
    m_OutputAssign -c A -o ".EDI" SYSPRINT
**m_OutputOverride**

**Name**

*m_OutputOverride* – Overrides an output file.

**Synopsis**

```
m_OutputOverride [-c class][-n copies][-d dest][-f forms][-H][-o list of output][-D dsname] -S Labelproc DDNAME
```

**Description**

This function overrides a sysout assignment (see *m_OutputAssign*).

**Options**

Only “-S” is mandatory, which specifies the step where the assignment is overridden. All other options are optional. See *m_OutputAssign* for other options.

**-S <label>**

Name of the label in the called procedure.
**m_OutputSet**

**Name**

*m_OutputSet* — manages the "OUTPUT JCL" statement with the following parameters: CLASS, COPIES, DEFAULT, DEST, FORMS, PRIORITY and WRITER.

It defines a reference and specifies associated processing options for sysout management.

**Synopsis**

```
m_OutputSet [-c][-n][-d][-f][-p][-w][-D] Reference
```

**Options**

- `-c <class>`
  
  Class of the output queue.

- `-n <copies>`
  
  Number of copies to print.

- `-d <dest>`
  
  Destination of the printing.

- `-f <forms>`
  
  Name of the form used.

- `-p <priority>`
  
  Specifies the priority of the output.

- `-w <external writer>`
  
  Specifies the use of an “external writer” to process the sysout file rather than TuxJes.

- `-D Y/N`
  
  Default reference (Y/N).

  Y indicates that the reference can be implicitly referenced in following m_OutputAssign function calls.

  N indicates that the reference can not be implicitly referenced in following m_OutputAssign function calls.

**Reference**

Reference name of the output. This name (and its associated characteristics) can be referenced in following m_OutputAssign function calls.
Example

m_OutputSet -d LOCAL -D N EDI
m_OutputAssign -c L -o "*.EDI" SYSUT2
m_PhaseBegin

Name
m_PhaseBegin — Called at the beginning of a script phase.

Synopsis
m_PhaseBegin

Description
This function is called at the beginning of a script phase.

Options
None.
m_PhaseEnd

Name
m_PhaseEnd — Called at the end of a script phase.

Synopsis
m_PhaseEnd

Description
This function is called at the end of a script phase.

Options
None.
**m_Pkzip**

*Name*

*m_Pkzip* — Executes zip process.

*Synopsis*

```
m_Pkzip [-f][-F] -a [-k][-n][-g][-d]
```

*Description*

This function zips files, directories or members of directory in an archive. Messages are reported in SYSPRINT file.

Note: the options “-f” and “-F” are not mandatory but at least one must be present.

*Options*

- **-f `<ddname1>,<ddname2>,<ddname3>...>`**
  
  First format of “-f” option, optional.

  ddname of the files to be zipped (enclosed with a single quote). Multiples ddname must be separated by a comma (,).

- **-f `<dirname>;<member1>...>`**
  
  Second format of “-f” option, optional.

  dirname followed by member names to be zipped (enclosed with a single quote).

  Member names must be separated by a semicolon (;).

  When dirname is alone, all the members are zipped.

  Wildcard (* or %) may be used (for member names only).

- **-F `<dsname1>,<dsname2>,<dsname3>...>`**
  
  Optional.

  dsname of the files to be zipped (enclosed with a single quote).

  Multiples dsname must be separated by a comma (,).

- **-a `<archive>`**
  
  mandatory.
ddname of the archive in which zipped file(s) are stored.

-k <action>
Optional.

Action to be done (ADD by default).

Use UPDATE to overwrite in archive.

-n ‘<zipped_name>’
Optional.

Names of the zipped file(s) in the archive (enclosed with single quotes) separated by a comma (,).

When several files are zipped, the same order than done in option “-f” followed by option”-F” must be respected.

-g <Y|N>
Optional.

GZIP compatible format.

Reserved for future used. This option is not taken into account by the RunTime Batch.

-d ‘<zipdir>’
Optional.

Directory (enclosed by single quotes) where files(s) to zip are located ($\{DATA\}$ by default).

Examples
In the following example, 2 files are to be zipped (ddname’s FICIN01 and FIC02) named by the “-f” option, DDARCH (ddname of the archive) is named by the ”-a” option while the first file will be renamed (see the “-n” option).

```
m_FileAssign -d SHR FICIN01 $(DATA)/PKZIP_FICIN01
m_FileAssign -d SHR FIC02 $(DATA)/FIC.FILE01
m_FileAssign -d NEW,CATLG DDARCH $(DATA)/ARCH.FILE
m_Pkzip -f ‘FICIN01,FIC02’ -a DDARCH -n ‘NEW_FICIN01,’
```
The following example shows how to use at the same time the options 

- `f 'ddn1,ddn2,ddn3,pds;member1;member2'`
- `F "dsn1,dsn2,dsn3'`
- `n ',newddn2,newwddn3,newmb1,,newdsn1,newdsn2,'`

The ddn1, member2 and dsn3 are not renamed (they are replaced by comma(,)).
m_Pkunzip

Name
m_Pkunzip — Executes unzip process.

Synopsis
m_Pkunzip [-f][-F] -a [-k][-n][-g][-d][-o][-w]

Description
This function unzips files, directories or members of directory in an archive.
Messages are reported in SYSPRINT file.
Note: the options “-f” and “-F” are not mandatory but at least one must be present.

Options
-f ‘<ddname1>[,<ddname2>[,<ddname3>]...]’
   Optional.
   ddname of the outfile(s) to be unzipped (enclosed with a single quote).
   Multiples ddname must be separated by a comma (,).
-F ‘<dsname1>[,<dsname2>[,<dsname3>]...]’
   Optional.
   dsname of the outfile(s) to be unzipped (enclosed with a single quote).
   Multiples dsname must be separated by a comma (,).
-a <archive>
   Mandatory.
   ddname of the archive from which zipped file(s) are extracted.
-o <outddname>
   Optional.
   ddname of the outfile intowhich file(s) are to be extracted.
-k <action>
   Optional.
Action to be done (EXTRACT by default).

Note: only EXTRACT is supported in this release.

-n ‘<zipped_name>’
Optional.

Names of the zipped file(s) in the archive (enclosed with single quotes) separated by a comma (,).

When several files are zipped, the same order than done in option “-f” followed by option”-F” must be respected.

-g <Y|N>
Optional.

GZUNZIP compatible format.

Reserved for future used. This option is not taken into account by the RunTime Batch.

-w <Y|N>
Optional.

Overwrite parameter (default N). Used to overwrite an existing file.

-d ‘<unzipdir>’
Optional.

Directory (enclosed by single quotes) where files(s) to unzip are to be stored (${DATA} by default).

Examples

Unzip all files of an archive without renaming files:

```
m_FileAssign -d NEW,CATLG DDARCHIV ${DATA}/PKZIP_ARCH
m_Pkunzip -a DDARCHIV
```

Unzip all files of an archive into an output file:

```
m_FileAssign -d NEW,CATLG DDARCHIV ${DATA}/PKZIP_ARCH
m_FileAssign -d NEW,CATLG OUTFILE ${DATA}/UNZIPPED.FILE
m_Pkunzip -a DDARCHIV -o OUTFILE
```
**m_ProcBegin**

**Name**

m_ProcBegin — Begins an in-stream procedure.

**Synopsis**

m_ProcBegin ProcedureName

**Description**

An in-stream procedure is added at the end of a korn shell script (by Oracle Tuxedo Application Runtime WorkBench during the translation) and referenced by m_ProcInclude.

**Options**

*ProcedureName*

Name of the procedure to include.

**Example**

    m_ProcBegin  KBPRB007
**m_ProcEnd**

**Name**

m_ProcEnd — Ends an in-stream procedure.

**Synopsis**

m_ProcEnd

**Description**

An in-stream procedure added at the end of a korn shell script is ended by m_ProcEnd.

**Options**

None.
**m_ProcInclude**

**Name**
m_ProcInclude — Calls a procedure to be included in the script during the conversion phase.

**Synopsis**
m_ProcInclude ProcedureName [param1=value1,param2=value2,…,paramN=valueN]

**Description**

**Options**

ProcedureName

Name of the (in-stream or catalogued) procedure to include.

**Example**
m_ProcInclude BPRAP001
m_ProgramExec

Name
m_ProgramExec — Executes a program.

Synopsis
m_ProgramExec [-b] [-e exit_type:exit_name][-n] Program [arguments]

Description
This function runs a COBOL program.

Options
-\(b\)
  Indicates the database will be accessed by the program.

-\(e\) exit_type:exit_name
  Indicates an exit routine should be used.

  An exit routine may be used at the beginning and/or at the end of a program.

  The exit type (BEGIN, END or BOTH) indicates if the exit routine must be called at the
  begin or at the end of the program (or both).

  The exit name is used to build the names of the sub-programs to be inserted before the call
  to the program (RTEX-"exitName"-Begin) and after the call of the program
  (RTEX-"exitName"-End).

-\(n\)
  This option is used to launch EXCI BATCH program. It informs the Batch Runtime to use
  runbexci but not runb.

  The environment variables used are (see RTBatch.conf file):
  - MT_EXCI
  - MT_JESDECRYPT
  - MT_EXCI_XA
  - MT_EXCIGRPNAME

  MT_EXCI is set to an non-empty string if EXCI interface exists and needs to use it,
  otherwise MT_EXCI is empty by default (needs Oracle Tuxedo).
In the last case, `MT_EXCI` must be set to `ART KIX` objects folder and `MT_JESDECRYPT` must be set to `jesdecrypt` object file (for example, /path/to/jesdecrypt.dynamic.linux64.o).

To use XA, `MT_EXCI_XA` must be set with the name of the resource manager for XA and `MT_EXCIGRPNAME` must be set with the `TUXEDO SRVGRP` value of `ARTDPL` server.

(For example, `MT_EXCI_XA=Oracle_XA` and `MT_EXCIGRPNAME=ResourceXA`)

If `MT_EXCI` is set to an non-empty string and `MT_EXCI_XA` and `MT_EXCIGRPNAME` are set to empty, you will connect to database without using XA.

During installation phase `MT_EXCI` is checked to know if the specific Runtime for EXCI should be linked.

During execution phase only `MT_EXCIGRPNAME` is checked.

Program [arguments]

Program name and user arguments to be passed to the program.

Note: When `m_ProgramExec` invokes nonexistent programs, JOB will continue if those programs are specified in `MTUTILITY_LIST_UNSUPPORT`.

Another way to use `runbexci` instead of `runb` is to configure the program in `MT_EXCI_PGM_LIST`. By doing this, whether or not `-n` is set, the program is invoked only by `runbexci`.

For more information about `MTUTILITY_LIST_UNSUPPORT` and `MT_EXCI_PGM_LIST`, please refer to the table of "Oracle Tuxedo Application Runtime for Batch Environment Variables (Optional)" in Using Batch Runtime.

Examples

```
m_ProgramExec BPRAB006 "08"
```

Indicates to run program BPRA006 with the parameter "08"

```
m_ProgramExec -b BDBAB001
```

Indicates that the program BDBAB001 accesses the Data Base

Note: To pass a parameter to a program

The "(" (double quote) character is used to mark out the boundaries of the parameters

Examples:

```
PARM=MT5 on z/OS becomes "MT5" on target
```
PARM=(MT5,MT6) on z/OS becomes "MT5,MT6" on target

PARM='S=MT5' on z/OS becomes "S=MT5" on target

PARM=(S=MT5,Q=MT6) on z/OS becomes "S=MT5,Q=MT6" on target

Two successive '"' (2 simple quotes) are replaced on one '"' (1 simple quote).

PARM='5 O’CLOCK' becomes "5 O’CLOCK"

Two successive "&&" (2 ampersands) are replaced by one "&" (1 ampersand) character.

'&&TEMP' becomes "&TEMP"

m_ProgramExec -e BEGIN:EX1 BPRAB006

Indicates to run program BPRA006 after the call of the “exit routine” RTEX-EX1-Begin.

This “exit routine” written by the user may contain user actions (for example concerning “accounting”).
**m_RcSet**

m_RcSet <ARGS> ReturnCode [Message]

**Name**
m_RcSet — Sets the return code.

**Synopsis**
m_RcSet ReturnCode [Message]

**Description**
m_RcTest sets the return code of a function.

**Options**

ReturnCode
   The value of the return code of the current phase.

Message
   A message that may be displayed with the return code.

**Examples**

```text
m_RcSet ${MT_RC_ABORT:-S999} "Unknown label : ${CURRENT_LABEL}"
```

```text
m_RcSet 0
```
m_SendMail

Name
m_SendMail — Sends an email.

Synopsis
m_SendMail -t [-f] [-s] [-m] [-a] [-n] [-c] [-b]

Description
This function sends an e-mail.

When the option “-f” is omitted, the environment variable MT_FROM_ADDRESS_MAIL must be declared in the user’s profile and initialized with the “From Address” to be used by default.

Two environment variables are used (see RTBatch.conf file):
MT_SMTP_SERVER : SmtpServer (default value : “localhost”)
MT_SMTP_PORT: SmtpPort (default value : “25”)

Options
-t ‘<To-Address1>[,<To-Adress2>...]’
Mandatory option. At least, one “To-Address” must be declared.
Electronic mail address of the recipient(s) inclosed in single quotes.
Multiple addresses must be separated by a comma “,”.

-f ‘<To-Address>’
Optional.
Electronic mail address of the sender inclosed in single quotes.
The environment variable MT_FROM_ADDRESS_MAIL is used when this option is missing.

-s “<Subject>”
Optional.
Subject of the email inclosed in double quotes.

-m <message file>
Optional.
ddname of the file containing the message of the email.

- \texttt{-a <attach-file1>,<attach-file2>...}
  Optional.

  ddname of the attached document(s).

  Multiple attachements must be separated by a comma “,“.

- \texttt{-n <filename1>,<filename2>...}
  Optional.

  Name of the attached document(s) referenced by option “-a”.

  Multiple names must be separated by a comma “,“.

  Note that there must have so many name as ddname.

- \texttt{-c <cc-Address1>,<cc-Address2>...}
  Optional.

  Electronic mail address of the Copy Carbone recipient(s) inclosed in single quotes.

  Multiple addresses must be separated by a comma “,“.

- \texttt{-b <bcc-Address1>,<bcc-Address2>...}
  Optional.

  Electronic mail address of the Blind Copy Carbone recipient(s) inclosed in single quotes.

  Multiple addresses must be separated by a comma “,“.

Example

In this example,
- the text of the message is stored in the sysin MESSAGE,
- the implicit “From-Address” is taken into the environment variable MT_FROM_ADDRESS_MAIL,
- the “To-Address” is ”BOB” <BOB.FOSTER@USA.COM>,
- an attached document is to be sent (with ddname file CR1 and name FILE2.CSV).
m_FileAssign -i MESSAGE

text of the email

_end

m_FileAssign -d SHR CR1 $\{DATA\}/FPROD.FILE1

m_SendMail -t ""BOB" <BOB.FOSTER@USA.COM>" -m MESSAGE -a CR1 -n 'FILE1.CSV'
m_ShellInclude

Name
m_ShellInclude — Inserts a part of script.

Synopsis
m_ShellInclude script name

Description
This function inserts a part of script.

Options

script name
Name of the part of a script to be included in the script shell during the conversion phase.
**m_Smtp**

**Name**

m_Smtp — Sends an email using SMTP protocol.

**Synopsis**

m_Smtp -i

**Description**

This function sends an e-mail using SMTP protocol. The SMTP commands are stored in the input file. The command uses the following format:

telnet <SmtpServer> <SmtpPort>

Two environment variables are used:

MT_SMTP_SERVER : SmtpServer (default value : “localhost”)

MT_SMTP_PORT: SmtpPort (default value : “25”)

**Options**

- **i <InputFile>**
  
  Mandatory option.
  
  ddname of the file containing the SMTP commands.

**Example**

m_Smtp -i SYSUT2
**m_StepLibSet**

**Name**

*m_StepLibSet* — Specifies where programs are stored.

**Synopsis**

```
m_StepLibSet directory [::directory[::directory...]]
```

**Description**

*m_StepLibSet* specifies at step level where programs are stored. This information is interpreted when the program is to be executed.

**Options**

**directory**

Path and name of the directory containing executable programs.
m_SymbolDefault

Name
m_SymbolDefault — Assigns a value to a symbol.

Synopsis
m_SymbolDefault var=value

Description
Used before the call of a procedure to define default substitution texts for symbols in the procedure.
This function will be analyzed and taken into account during the conversion phase and the symbols replaced by their value in the extended script.

Options
var
Name of the variable.

Value
Value assigned to the variable.

Example
m_SymbolDefault VAR=45
m_SymbolSet

Name
m_SymbolSet — Defines a symbol.

Synopsis
m_SymbolSet var=value

Description
Defines a symbol and assigns a value before the first use of this symbol.

Options
var
Name of the variable.

Value
Value assigned to the variable.

Example
m_SymbolSet VAR=45
TVAL=$(date +%Y%m%d) (value of command)
m_SymbolSet TVAR=$TVAL
**m_UtilityExec**

**Name**
m_UtilityExec — Executes the stored commands.

**Synopsis**
m_UtilityExec [sysin file]

**Description**
Executes the contents of an external sysin. This function executes the script “UtilityName” in the current Shell. UtilityName is the physical name previously assigned to [sysin file].

The script assigned with sysin should contain a sequence of RunTime Batch functions. Although this function can also launch native Korn shell scripts, it is recommended to use m_ProgramExec instead. In this way, assignation is not required.

**Options**

**sysin file:**
By default SYSIN.

ddbname of a sysin file which contains function to launch.

**Example**

m_FileAssign -d NEW SYTSIN ${SYSIN}/SYSTIN/MUEX07.sysin

m_UtilityExec SYTSIN

The file MUEX07.sysin may have the following content:
m_ProgramExec -b ZVDL101

**m_SetJobExecLocation**

**Name**
m_JobSetExecLocation — Specify the JES Tuxedo server group, in which the current job is executed.

**Synopsis**
m_JobSetExecLocation <SvrGrpName>
Description
Specify the JES Tuxedo server group, in which the current job is executed.

This API is optional; once it is used, it should immediately follow \texttt{m\_JobBegin}.

If this API is not called in a job, the job will be randomly executed on any server group; on the other hand, if this API is called in job, the job will be executed on the specified server group. If the specified server group does not exist or no \texttt{ARTJESINITIATOR} in that server group is started, the job will be in the waiting queue until the \texttt{ARTJESINITIATOR} in that server group is started.

If \texttt{SvrGrpName} is not specified or "" is specified, although no errors are reported, the job will be randomly executed on any server group.

Options
\textbf{SvrGrpName}
Mandatory.
The JES server group name. The \texttt{ARTJESINITIATOR} deployed in this server group executes the current job.

Example
\begin{verbatim}
m\_JobBegin -j SAMPLEJCL -s START -v 2.0 -c R
m\_JobSetExecLocation "ATLANTA"
   while true ;
   do
      m\_PhaseBegin
   ...

m\_JobEnd
\end{verbatim}

The job can be submitted on any JES node, but only be executed by the \texttt{ARTJESINITIATOR} which belongs to JES's Tuxedo server group \texttt{ATLANTA}.
Tuxedo Job Enqueueing Service (TuxJES)

This chapter contains the following topics:

TuxJES Servers, Commands, and Utilities
TuxJES Job Operation

**TuxJES Servers, Commands, and Utilities**

This chapter describes servers, commands, and utilities included in the TuxJES feature. 

**Table 1** lists TuxJES commands and functions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>genjesprofile</td>
<td>Generates the security profile for TuxJES system</td>
</tr>
<tr>
<td>artjesadmin</td>
<td>TuxJES command interface.</td>
</tr>
<tr>
<td>artjescleanlock</td>
<td>Cleans stale lock records in lock files in MP mode.</td>
</tr>
<tr>
<td>ARTJESADM</td>
<td>TuxJES administration server.</td>
</tr>
<tr>
<td>ARTJESCONV</td>
<td>TuxJES conversion server.</td>
</tr>
<tr>
<td>ARTJESINITIATOR</td>
<td>TuxJES job control API.</td>
</tr>
<tr>
<td>ARTJESPURGE</td>
<td>Purges job queue.</td>
</tr>
</tbody>
</table>
**genjesprofile**

**Name**

*genjesprofile* – Generates the security profile for TuxJES system

**Synopsis**

`genjesprofile [-f userprofile]`

**Description**

This utility generates the security profile for TuxJES system. When `genjesprofile` is launched, you are prompted to enter the Oracle Tuxedo application password, user name, user password and the database connection (`MT_DB_LOGIN`), and the database login information to access Oracle Database for GDG management (`MT_GDG_DB_ACCESS`). The output is a security profile file which contains the Oracle Tuxedo application password, user name, user password and the database connection with the file permission “-rw-------”.

**Note:** The generated security profile file can be used by the *artjesadmin* tool to login to an Oracle Tuxedo domain.

**Parameters and Options**

`genjesprofile` supports the following parameters and options:

`[-f <output_file>]`

The location of the generated security profile file. If this option is not specified, the default value is `~/.tuxAppProfile`.

**See Also**

*artjesadmin*
artjesadmin

Name

artjesadmin – TuxJES command interface.

Synopsis

- artjesadmin -v
- artjesadmin [-f [security_profile]]
- artjesadmin [-f [security_profile]] -p -j jobid
- artjesadmin [-f [security_profile]] [-o ejr_option] [-s shell_option] [-y [-t timeout(s)]] -i/-I scriptfile

Description

artjesadmin is the TuxJES command interface. It requires the TuxJES system must be started first.

Parameters and Options

artjesadmin supports the following parameters and options:

- **-h**
  Option -h prints artjesadmin usage information as listed above.

- **-v**
  Option -v indicates the current version of Oracle Tuxedo Application Runtime for Batch.

- **-i**
  Option -i is specified to submit KSH jobs. scriptfile should be full path for one KSH job.

- **-I**
  Option -I is specified to submit JCL jobs. scriptfile should be full path for one JCL job.

- **-f**
  The security profile file generated by genjesprofile. The default value is ~/.tuxAppProfile. It requires the owner of the security profile must be the user who runs artjesadmin. The user name in this profile is the owner of the submitted jobs. A job without a specified owner is assigned the owner name "*".
A job with a particular owner can only be controlled by that owner. A job without a particular owner (*) can be controlled by anyone. Any user can print all jobs.

**Note:**
- If `-f profile` is specified, the specified profile file will be used;
- If `-f` is specified while profile is not, `~/.tuxAppProfile` will be used as the profile file;
- If `-f` is not specified, nothing will be used as the profile file.

**-y and -t**

**Descriptions**

Option `-y` and `-t` are added to submit a job in the synchronous way. **Table 2** shows some details.

**Table 2 Option -y and -t Descriptions**

<table>
<thead>
<tr>
<th>Option</th>
<th>Value Range</th>
<th>Descriptions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-y</td>
<td>N/A</td>
<td>Enables synchronous mode to wait for job end.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
| -t     | 1 ~ (2^31 -1)   | Specifies the timeout value.         | Optional. If `-t` is omitted, `artjesadmin` will wait infinitely.
|        |                 |                                     | While timeout occurs, `artjesadmin` command line will exit but the job will run by JES continuously without impact. |

**Note:** To submit a job in synchronous way, in TuxJES Configuration File, it's required to set `EVENTPOST=A`; in UBBCONFIG file, it’s required to set `NOTIFY` to `DIPIN` and set server `TMUSREVT`.

**Exit code**

**Table 3** lists the exit codes for `artjesadmin` if `-y` is specified.
Table 3 Exit Code

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Descriptions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Job is finished successfully.</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>Command execution fails.</td>
<td>The code will be returned by either invalid timeout value or a command line syntax error.</td>
</tr>
<tr>
<td>2</td>
<td>Job conversion fails.</td>
<td>Job is submitted successfully, but job conversion fails.</td>
</tr>
<tr>
<td>3</td>
<td>Job execution fails.</td>
<td>Job is submitted successfully, but job execution fails.</td>
</tr>
<tr>
<td>4</td>
<td>Job is canceled.</td>
<td>Job is submitted successfully, but is canceled before reaching final status.</td>
</tr>
<tr>
<td>20</td>
<td>Timeout occurs.</td>
<td>Job has not been finished within the time specified by -t option.</td>
</tr>
</tbody>
</table>

Standard Output

Information shown on Table 4 will be printed to stdout in the following format.

<JOBID>,<JOBNAME>,<JOBSTATUS>,<JOB RETURN CODE>

Table 4 Standard Output

<table>
<thead>
<tr>
<th>Output Content</th>
<th>Descriptions</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;JOBID&gt;</td>
<td>Job ID</td>
<td>00005097</td>
</tr>
<tr>
<td>&lt;JOBNAME&gt;</td>
<td>Job name</td>
<td>JOBA</td>
</tr>
<tr>
<td>&lt;JOBSTATUS&gt;</td>
<td>Job final status (only available if job is finished before the timeout occurs)</td>
<td>DONE</td>
</tr>
<tr>
<td>&lt;JOB RETURN CODE&gt;</td>
<td>Job return code from EJR (only available if a job is finished before the timeout occurs)</td>
<td>C000</td>
</tr>
</tbody>
</table>

Listing 1  Sample: Job is Executed Successfully

00000002,JOBA,DONE,C0000
Listing 2  Sample: Job Fails

00000002,JOBA,FAILED,U0568

Listing 3  Sample: Timeout Occurs

00000002,JOBA,Already Timeout!

-p and -j

Descriptions
Option -p and -j are added to get job status without interaction in artjesadmin console.

Exit Code
Table 5 lists the exit codes for artjesadmin if -p is specified.

Table 5  Exit Code

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Descriptions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Job is finished normally. Job status = DONE</td>
<td>A job is finished successfully.</td>
</tr>
<tr>
<td>1</td>
<td>Command execution fails.</td>
<td>The failure is caused by an internal error, a network error, or a syntax error.</td>
</tr>
<tr>
<td>3</td>
<td>Job status = FAIL</td>
<td>JOB execution fails.</td>
</tr>
<tr>
<td>4</td>
<td>Job status = CANCEL</td>
<td>A job is canceled.</td>
</tr>
<tr>
<td>5</td>
<td>Job status = CONVING</td>
<td>A job is waiting for conversion.</td>
</tr>
<tr>
<td>6</td>
<td>Job status = EXECUTING</td>
<td>A job is running.</td>
</tr>
<tr>
<td>7</td>
<td>Job status = HOLD_CONVING</td>
<td>A job is in hold state without conversion.</td>
</tr>
<tr>
<td>8</td>
<td>Job status = HOLD_WAITING</td>
<td>A job is in hold state after conversion.</td>
</tr>
</tbody>
</table>
**Standard Output**

Information shown on Table 6 will be printed to stdout in the following format.

\(<JOBID>,<JOBNAME>,<JOBSTATUS>,<JOB RETURN CODE>\)

---

**Table 5 Exit Code**

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Descriptions</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Job status = WAITING</td>
<td>A job is waiting for execution.</td>
</tr>
<tr>
<td>10</td>
<td>Job status = DISCARD</td>
<td>This status will occur if tpenqueue() fails.</td>
</tr>
<tr>
<td>11</td>
<td>Job status = INDOUBT</td>
<td>When a job is running, if JES server ARTJESINITIATOR is shutdown and then restarted, the job status will be INDOUBT.</td>
</tr>
<tr>
<td>22</td>
<td>Job doesn’t exist.</td>
<td>N/A</td>
</tr>
</tbody>
</table>

---

**Table 6 Standard Output**

<table>
<thead>
<tr>
<th>Output Content</th>
<th>Descriptions</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;JOBID&gt;</td>
<td>Job ID</td>
<td>00005097</td>
</tr>
<tr>
<td>&lt;JOBNAME&gt;</td>
<td>Job name</td>
<td>JOBA</td>
</tr>
<tr>
<td>&lt;JOBSTATUS&gt;</td>
<td>Job current status</td>
<td>DONE</td>
</tr>
<tr>
<td>&lt;JOB RETURN CODE&gt;</td>
<td>Job return code from EJR (only available if a job has finished)</td>
<td>C000</td>
</tr>
</tbody>
</table>

---

**Listing 4**  Sample: Job has been Finished Normally

00000002,JOBA,DONE,C0000

---

**Listing 5**  Sample: Job is Finished but Fails

00000002,JOBA,FAILED,U0568
Listing 6  Sample: Job is Running

00000002,JOBA,EXECUTING

artjesadmin supports the following sub commands:

```
submitjob(smj )[-o='xxx'] -i/-I scriptfile
```

Submits a job to TuxJES system. The `scriptfile` parameter is the job script to be submitted.

**Note:** `artjesadmin` is not responsible for scriptfile propagation. It must be located on a shared file system if the conversion and execution are not on same machines. The options are as follows:

- `-o='xxx'`: Specifies the options passed to the EJR script file.
- `-i =scriptfile`: The script file.
- `-I`: The option specified to submit JCL jobs.

Once successfully invoked, the return format `Job xxx is submitted successfully. If an error occurs, an error message is printed.

`artjesadmin` also supports direct job submission using the following format:

```
artjesadmin [-o='xxx'] -i/-I scriptfile.
```

`artjesadmin` has a return code different from zero if there is an error occurs as listed in Table 7.

### Table 7  Error Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No runtime error</td>
</tr>
<tr>
<td>251</td>
<td><code>artjesadmin</code> it self command error returned by ARTJESADM server</td>
</tr>
<tr>
<td>252</td>
<td>JES2SUBMIT service error</td>
</tr>
<tr>
<td>Others</td>
<td>EJR none zero exit code</td>
</tr>
</tbody>
</table>
printjob(ptj) -n jobname | -j jobid | -c job_class | -a [-v][-t JCL|KSH] [-m]

Displays the existing jobs. If no option is specified, it displays all jobs. The options are as follows:

- `-n jobname`: Display jobs with given job name
- `-j jobid`: Display a particular job information
- `-c job_class`: Display a particular class jobs information
- `-a`: Display all jobs
- `-v`: Verbose mode
- `-t`: JCL|KSH: Display JCL|KSH jobs
- `-m`: Print the CPU time usage of each step in one JOB

**Listing 7  printjob Output**

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>JobID</th>
<th>Owner</th>
<th>Prty</th>
<th>C</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>cjob</td>
<td>00000015</td>
<td>*</td>
<td>5</td>
<td>A</td>
<td>KSH</td>
<td>DONE</td>
</tr>
<tr>
<td>cjob</td>
<td>00000016</td>
<td>*</td>
<td>5</td>
<td>A</td>
<td>KSH</td>
<td>DONE</td>
</tr>
<tr>
<td>cjob</td>
<td>00000018</td>
<td>*</td>
<td>5</td>
<td>A</td>
<td>KSH</td>
<td>CONVING</td>
</tr>
</tbody>
</table>

*total: 3*
*success: 3*

- **JOBNAME**: The job name.
- **JobID**: The job ID generated by TuxJES system
- **Type**: the job type (JCL or KSH)
- **Owner**: Job Owner.
- **Prty**: Job priority
- **C**: The job class.
- **Status**: Job status
- **EXECUTING**: a job is running
- **CONVING**: a job waiting for conversion
- **WAITING**: a job waiting for execution.
- **DONE**: a job finished successfully.
- **FAIL**: a job finished but failed
- **HOLD_WAITING**: a JOB is in hold state after conversion
- **HOLD_CONVING**: a job is in hold state without conversion
- **INDOUBT**: a job is in doubt state due to its initiator restarted
- **CANCELED**: a job is canceled

In verbose mode, the job detail information is displayed:

- Submit time: The submit time of the job
- Step: The current running job step. It is only applicable to running jobs.
- Type Run: The TYPRUN definition of the job.
- Machine: Only for running/done/failed jobs. It is the machine name that the job is/was running on.
- CPU usage: The user CPU usage and system CPU usage for the job execution.
- Result: Job operation result, “OK” or error message.

If no option is specified, the "-a" option is assumed.

```
holdjob(hj) -n job name | -j jobid | -c job_class | -a
```

Hold the specified jobs which are in \texttt{CONVING} or \texttt{WAITING} status. The options are as follows:

- \texttt{-n jobname}: Hold jobs with given job name
- \texttt{-j jobid}: Hold a particular job
- \texttt{-c job_class}: Hold a particular class jobs
- \texttt{-a}: Hold all jobs

If no option is specified, the "-a" option is assumed.
releasejob(rlj) -n job name | -j jobid | -c job_class | -a
Releases the jobs in HOLD_WAITING or HOLD_CONVING status so that they can be picked up by ARTJESCONV for conversion or ARTJESINITIATOR for running. The options are as follows:
- n  jobname: Release jobs with given job name
- j  jobid: Release a particular job
- c  job_class: Release a particular class jobs
- a: Release all jobs

If no option is specified, the "-a" option is assumed.

canceljob(cj) -n job name | -j jobid | -c job_class l -a
Cancels a job and moves it to the output queue. For running jobs, this command informs the related ARTJESINITIATOR to invoke EJR with "-k" option. Other jobs are moved directly to the output queue. The TuxJES system assumes the job is terminated when EJR returns. The options are as follows:
- n  jobname: Cancel jobs with given job name
- j  jobid: Cancel a particular job
- c  job_class: Cancel a particular class jobs
- a: Cancel all jobs

If no option is specified, the "-a" option is assumed.

purgejob(pgj) -n job name | -j jobid | -a
Completed jobs in the output queue are moved to the purge queue. For other jobs, purgejob has same effect as canceljob. The purgejob command does not purge the job directly. The ARTJESPURGE server deletes the job from the TuxJES system. If ARTJESPURGE is not started, the job remains in the output queue.
The options are as follows:
- n  jobname: Purge jobs with given job name
- j  jobid: Purge a particular job
- a: Purge all jobs

If no option is specified, the "-a" option is assumed.

changeconcurrent(chco) -g groupname -i serverid -n concurrent_num
Changes the number of maximum concurrent executing jobs for the ARTJESINITIATOR server which is designated by the -g and -i options. The change takes effect with no need to restart the ARTJESINITIATOR server.
The options are as follows:

- **-g groupname**: the Tuxedo group name of the ARTJESINITIATOR server
- **-i serverid**: the Tuxedo server id of the ARTJESINITIATOR server
- **-n concurrent_num**: the number of maximum concurrent executing jobs

The change is not persistent, which means the number is reset when the ARTJESINITIATOR server restarts.

```
printconcurrent(pco) -g groupname -i serverid
```

Displays the number of maximum concurrent executing jobs for the ARTJESINITIATOR server which is designated by -g and -i options.

The options are as follows:

- **-g groupname**: the Tuxedo group name of the ARTJESINITIATOR server
- **-i serverid**: the Tuxedo server id of the ARTJESINITIATOR server

```
event (et) [-t C,E,P,L,A] on|off
```

This command tells artjesadmin to subscribe particular job event. The options are:

- **C**: job conversion complete event; the event name is `ARTJES_JOBCVT`
- **E**: job execution finish event; the event name is `ARTJES_JOBEXEC`
- **P**: job purge event; the event name is `ARTJES_ARTJESPURGE`
- **L**: job cancel completed event; the event name is `ARTJES_JOBCANCEL`
- **A**: all supported events. If the event is set to "on", A is the default.

```
on |off: The submission is on or off. The "on" setting can be used with the -t option. "off" will unsubscribe all event subscriptions.
```

If the subscribed event type is not configured in JESCONFIG, an error is reported.

```
verbose(v) on|off
```

Turn on/off verbose mode.

See Also

Oracle Tuxedo Application Runtime for Batch User Guide

**artjescleanlock**

Name

artjescleanlock – The utility to clean stale lock records in lock files in MP mode.
Synopsis

artjescleanlock -c [-y] [-h <hostname>] [-f <filefullpath>]
artjescleanlock -l [-h <hostname>] [-f <filefullpath>]

Description

Artjescleanlock, based on the filter criteria specified by its options, is used to list or clean lock records in lock files. Before running this utility, environment variable $MT_ACC_FILEPATH$ must be set properly.

Parameters and Options

artjescleanlock supports following options:

- `-l`
  Specifies the users want to just list or display the lock records matching the specified filter criteria.

- `-c`
  Specifies the users want to clean the lock records matching the specified filter criteria.

Note: One and only one of “-l” and “-c” must be set; they are exclusive to each other.

[ `-h hostname` ]
  Optional: It specifies a hostname as the filter criteria. Only lock records matching such hostname are impacted.

[ `-f filename` ]
  Optional: It specifies a filename as the filter criteria. Only lock records matching this filename are impacted.

Note: If no filter criteria are supported, all lock records will be impacted.

[ `-y` ]
  Optional: It specifies whether the users want to forcibly clean the lock records without confirmation. Without “-y”, the utility will ask for users’ confirmation before cleaning. This option can only be used with “-c”.

Name

ARTJESADM – TuxJES Administration server.
Tuxedo Job Enqueueing Service (TuxJES)

Synopsis

ARTJESADM
SRVGRP="identifier"
SRVID="number" CLOPT="[-A][servopts options] -- -i JESCONFIG"

Description

ARTJESADM is an Oracle Tuxedo application server provided by TuxJES. The artjesadmin command communicates with ARTJESADM for most tasks.

ARTJESADM must be configured in the UBBCONFIG file in front of other TuxJES servers since others they access services provided by ARTJESADM. If JESCONFIG is changed, all TuxJES related servers must be restarted for new configurations to take effect.

Parameters and Options

ARTJESADM supports the following parameters and options:

- **i JESCONFIG**
  JESCONFIG represents the full path name of the TuxJES system configuration file. It allows the following parameters:

  **JESROOT**
  The full path name of the root directory to store job information. It is a mandatory attribute. If this directory does not exist, ARTJESADM creates it automatically.
  
  JESROOT=/xxx/xxx

  **DEFAULTJOBCLASS**
  The default job class if the job class is not set for a job. It is an optional attribute. The default job class is A if this attribute is not set.
  
  DEFAULTJOBCLASS=[A-Z],[0=9]

  **DEFAULTJOBPRIORITY**
  The default job priority if the job priority is not set for a job. It is an optional attribute. The default job priority is 0 if this attribute is not set.
  
  DEFAULTJOBPRIORITY=[0-15]

  **DUPL_JOB=NODELAY**
  If it is not set, only one job can be in execution status for a job name. NODELAY will remove the dependency check. The default value is delay execution.

  EVENTPOST=S,C,E,P,L,A
  Specifies whether events are posted for a job at particular stages.

  S: Job submission event. Event name: ARTJES_JOBSUBMIT
C: Job conversion complete event. Event name: ARTJES_JOBCVT
E: Job execution finish event. Event name: ARTJES_JOBFINISH
P: Job purge event. Event name: ARTJES_JOBPURGE
L: Job cancel completed event. Event Name: ARTJES_JOBCANCEL
A: All supported events.

If EVENTPOST is not specified, no events are posted. The data buffer with event pos is FML32 type and the fields are defined in JESDIR/include/jesflds.h.

JOBREPOSITORY
The path of the job repository where jobs are stored. The script file path inputted in job submitting may be a relative path in JOBREPOSITORY if it is set.

PRIVILEGE_MODE
Specifies whether and how to enable the user substitution (For more information, see JuxJES User Substitution in Oracle Tuxedo Application Runtime for Batch Reference Guide). The values are:

NONE: Default value. Indicates jobs are executed by the OS user who starts JES system. This is compatible with all previous implementations on JES system.

USER_IDENTICAL: Indicates jobs are executed by the Oracle Tuxedo user with which JES client joins JES system. Make sure that each Oracle Tuxedo user corresponds to an existing OS user before you choose this value.

USER_MAPPING: When this value is specified, the JES system looks up the TuxJES user mapping file and finds out the OS user corresponding to the Oracle Tuxedo user with which JES client joins JES system, and then appoints this OS user as the job executor.

USER_MAPPING_FILE
The full path where TuxJES user mapping file is stored. It is used along with PRIVILEGE_MODE when its value is USER_MAPPING.

It is recommended that the owner of user mapping file is root and the file permission is "-rw-------".

One Oracle Tuxedo user must be mapped to one OS user only, and it is recommended that one OS user is mapped to one Tuxedo user as well.

Example(s)
UBBCONFIG example:

ARTJESADM
SRVID=1 SRVGRP=SYSGRP CLOPT="-A -- -i /nfs/users/jes/jesconfig"
Tuxedo Job Enqueueing Service (TuxJES)

See Also

Oracle Tuxedo Application Runtime for Batch User Guide

ARTJESCONV

Name

ARTJESCONV – TuxJES conversion server.

Synopsis

ARTJESCONV
SRVGRP="identifier"
SRVID="number" CLOPT=" [A][servopts options] -- "

Description

The TuxJES conversion server. It is responsible for invoking the EJR to do the job conversion.

Example(s)

UBBCONFIG example:

ARTJESCONV

SRVID=2 SRVGRP=SYSGRP CLOPT="-A -- "

See Also

Oracle Tuxedo Application Runtime for Batch User Guide

ARTJESINITIATOR

Name

ARTJESINITIATOR – Job Initiator

Synopsis

ARTJESINITIATOR
SRVGRP="identifier"
SRVID="number" CLOPT=" [A][servopts options] -- -C jobclasses [-n concurrent_num] [-d]"
Description

ARTJESINITIATOR is an Oracle Tuxedo application server provided the TuxJES. It is responsible for invoking the EJR to execute the jobs.

Once a ARTJESINITIATOR is killed or shutdown while it has job running, it will put the job in the INDOUBT state when it is restarted.

Parameters and Options

ARTJESINITIATOR supports the following parameters and options:

-c jobclasses[jobclass]
  Specifies the job classes this ARTJESINITIATOR server is associated. If this option is not specified, ARTJESINITIATOR associates with all job classes.

-n concurrent_num
  Specifies the number of maximum concurrent executing jobs for this ARTJESINITIATOR server. The default value is 1.

-d
  Specifies the number of maximum concurrent executing jobs for this ARTJESINITIATOR server can be change by artjesadmin changeconcurrent command.

Example(s)

UBBCONFIG example:

ARTJESINITIATOR

  SRVID=3 SRVGRP=SYSGRP MIN=10 CLOPT="-A -- -c AHZ"

In this example, ten ARTJESINITIATOR instances are configured and are associated with the "A", "H" and "Z" job classes.

See Also

Oracle Tuxedo Application Runtime for Batch User Guide

ARTJESPURGE

Name

ARTJESPURGE — Purges job queue

Synopsis

ARTJESPURGE
SRVGRP="identifier"
SRVID="number" CLOPT=" [-A][servopts options] -- "

Description
ARTJESPURGE monitors the purge queue. If it finds a job in the purge queue, it removes the job in the queue and deletes the directory JESROOT/<JOBID>.

See Also
Oracle Tuxedo Application Runtime for Batch User Guide

TuxJES Queue System
In order to emulate the z/OS JES2 system, TuxJES system uses a queue mechanism for batch job life cycle management. All queues are created in one queue space called "JES2QSPACE". A batch job is represented by a message that resides and is transferred to queues listed in Table 8.

Table 8  TuxJES Queues

<table>
<thead>
<tr>
<th>Queues</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Conversion Queue</td>
<td>When a JCL job is submitted to the TuxJES system, it is put in the JCL conversion queue first. There is only one JCL conversion queue in the system. A JCL converted job is moved from the &quot;JCL conversion queue&quot; to the &quot;conversion queue&quot;. The jobs in the queue are processed in FIFO order. Queue name: CONV_JCL</td>
</tr>
<tr>
<td>Conversion Queue</td>
<td>When a batch job is submitted to the TuxJES system, it is put in the conversion queue first. There is only one conversion queue in the system. A converted job is moved from the &quot;conversion queue&quot; to the &quot;execution queue&quot;. The jobs in the queue are processed in FIFO order. Queue name: CONV</td>
</tr>
<tr>
<td>ExecGroup Queue</td>
<td>This queue is necessary if NJESUPPORT is enabled in jesconfig file. This queue maintains the mapping of each job and its execution group when the job is in the Execution Queue. Queue name: EXECGRP</td>
</tr>
</tbody>
</table>
The TuxJES Queue System

The TuxJES Queue System provides a sample shell script (jesqinit) to create the queue space (JES2QSPACE) and the queues listed in Table 8. You can modify the script to adapt to your environment, but must adhere to the following:

1. Queue order can not be changed
2. Fixed queue names and queue space name
3. The script can be customized for queue space/queue creation parameters

Table 8 TuxJES Queues

<table>
<thead>
<tr>
<th>Queues</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold Queue</td>
<td>If a job is in the HOLD state (JCLHOLD or HOLD), it is put in the hold queue. Once released, it is moved to the conversion queue or waiting queue based on the typrun parameter. Queue name: HOLD</td>
</tr>
<tr>
<td>Execution Queue</td>
<td>There are 36 job classes (A-Z and 0-9). A job also has a priority value ranging from 0 to 15. The jobs are scheduled based on the job class and priority. One job class is mapped to one /Q queue, (36 queues all together). These are the queues where the job is stored staying and waits for execution. The job priority is mapped to the queue message priority. All queues are created based on priority. Queue names: [A-Z], [0-9].</td>
</tr>
<tr>
<td>Executing Queue</td>
<td>This queue stores running/executing jobs. There is only one &quot;executing queue&quot; in the system. When a job is picked up from an &quot;execution queue&quot; and successfully goes to running state, the job is moved to the &quot;executing queue&quot;. The jobs in this queue are processed in FIFO order. Queue name: EXEC</td>
</tr>
<tr>
<td>Output Queue</td>
<td>When a job is completed or an error occurs, it is sent to the &quot;output queue&quot;. There is only one &quot;output queue&quot;. The jobs in the queue are processed in FIFO order. Queue name: OUTPUT</td>
</tr>
<tr>
<td>Purge Queue</td>
<td>When a job is to be purged, it is moved to the purge queue. There is only one &quot;purge queue&quot; in the system. Queue name: PURGE</td>
</tr>
<tr>
<td>Internal Queues</td>
<td>ART JES also has some internal queues on the JES2QSPACE for internal usage.</td>
</tr>
</tbody>
</table>
Recommended /Q Creation Values

Device Size of Pages: 10000
Queue Space Size of Pages: 5000 (We assume the max number of jobs is 10000, each job will consume 2k bytes and the page size is 4k)
Number of Messages in Queue: 10000
Number of Concurrent Transactions: 1000
Number of Concurrent processes in queue: 100

Note: These parameters can be customized according to the specific environment.

TuxJES User Substitution

JES supports defining the specific OS user to execute jobs. With the Tuxedo User-OS User mapping mechanism, JES system is able to associate the Tuxedo AAA credential adhering to a job with an OS user who has the limited permission to access resources when operating a job.

Terms

Table 9 lists the terms you need to know for a good understanding of the user substitution feature described in this section.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job submitter</td>
<td>The OS user who submits a JES job with artjesadmin</td>
</tr>
<tr>
<td>Job owner</td>
<td>The Oracle Tuxedo user who is displayed as the owner of a JES job</td>
</tr>
<tr>
<td>Job user</td>
<td>The OS user who submits, cancels or prints a JES job with artjesadmin</td>
</tr>
<tr>
<td>Job executor</td>
<td>The OS user who executes the EJR to execute a JES job</td>
</tr>
</tbody>
</table>

Job Owner Designation in Different Scenarios

The Job owner designated by JES system depends on the combination of configurations on artjesadmin and PRIVILEGE_MODE. Table 10 shows the relation between job owner designation and different configuration scenarios.
Permissions of Job Operations

Table 11 and Table 12 show the permissions of users with different identities when operating jobs in TuxJES system enabling the user substitution.

**Table 10 Job Owner Designation in Different Configuration Scenarios**

<table>
<thead>
<tr>
<th>Configuration Scenario</th>
<th>Job Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When artjesadmin is executed with...</strong></td>
<td><strong>When PRIVILEGE_MODE is set to...</strong></td>
</tr>
<tr>
<td>a profile containing a user name</td>
<td>NONE</td>
</tr>
<tr>
<td>a profile containing a user name</td>
<td>USER_IDENTICAL or USER_MAPPING</td>
</tr>
<tr>
<td>no profile or a profile containing no user name</td>
<td>NONE</td>
</tr>
<tr>
<td>no profile or a profile containing no user name</td>
<td>USER_IDENTICAL or USER_MAPPING</td>
</tr>
</tbody>
</table>

**Note:** Oracle Tuxedo SECURITY parameter must be set to USER_AUTH, ACL or MANDATORY_ACL, otherwise JES system fails to start up and the following error message is printed into ULOG: "ERROR: The current Tuxedo security level disallow the given privilege mode".

A job with an owner "*" can be controlled by anyone.
Enabling User Substitution in TuxJES

To enable the user substitution in TuxJES, the following operations should be done:

- Configuring JESCONFIG in TuxJES
- Booting Up JES System by Root User
- Configuring Tuxedo UBBCONFIG
- Configuring File System Permission

Configuring JESCONFIG in TuxJES

The value of PRIVILEGE_MODE in JESCONFIG should be specified to MAPPING_CREDENTIAL or IDENTITY_CREDENTIAL.
If the value of PRIVILEGE_MODE in JESCONFIG is MAPPING_CREDENTIAL, the value of USER_MAPPING_FILE should be specified and the user mapping file should contain the mapping between Oracle Tuxedo users and OS users.

**Booting Up JES System by Root User**

When JES is working in SHM mode, it should be booted up by root user.

When JES is working in MP mode, do one of the following:

- Execute `tmboot` by root when JES system is running on the master machine of a MP domain.
- Execute `tlisten` by root when JES system is running on a slave machine of a MP environment.

It is recommended that all machines involved in a JES system use the same NIS system.

**Configuring Tuxedo UBBCONFIG**

All job users should have full permissions to IPC resource of the JES system.

The SECURITY parameter in RESOURCE section of TUXCONFIG should be USER_AUTH, ACL or MANDATORY_ACL.

**Note:** If SECURITY parameter is set to APP_PW and AUTHSVC parameter is configured, the SECURITY level is regarded as USER_AUTH by Oracle Tuxedo.

Here is an example of UBBCONFIG:

**Listing 8  Oracle Tuxedo  UBBCONFIG File Example When Enabling User Substitution in TuxJES System**

```plaintext
# (c) 2011 Oracle. All Rights Reserved.
#
*RESOURCES
IPCKEY <IPCKEY> # for example 132770
DOMAINID jessample
MASTER SITE1
MODEL SHM
MAXACCESSERS 200
```
MAXSERVERS  50
NOTIFY     SIGNAL
PERM       0666 #Adding "PERM=0666" in RESOURCES section
SECURITY   USER_AUTH
AUTHSVC    "AUTHSVC"

*MACHINES
#
<uname -n>

   LMID = SITE1
   TUXDIR ="<full path of TUXEDO software>"
   TUXCONFIG = "<full path of APPDIR>/tuxconfig"
   TLOGDEVICE ="<full path of APPDIR>/TLOG"
   TLOGSIZE=10
   APPDIR = "<full path of APPDIR>"
   ULOGPFX = "<full path of APPDIR>/ULOG"

*GROUPS

ARTGRP
   LMID = SITE1  GRPNO = 1

QUEGRP
   LMID = SITE1  GRPNO = 2
   TMSNAME = TMS_QM  TMSCOUNT = 2
   OPENINFO = "TUXEDO/QM:<full path of APPDIR>/QUE:JES2QSPACE"
EVTGRP
   LMID= SITE1 GRPNO=3

# *SERVERS
# Adding RQPERM=0666 RPPERM=0666 in all JES servers entry in SERVERS section
DEFAULT: CLOPT="-A"

TMUSREVT          SRVGRP=EVTGRP SRVID=1 CLOPT="-A"
                   RQPERM=0666 RPPERM=0666

TMQUEUE
   SRVGRP = QUEGRP SRVID = 1
   GRACE = 0 RESTART = Y CONV = N MAXGEN=10
   CLOPT = "-s JES2QSPACE:TMQUEUE -- -t 5 "
   RQPERM=0666 RPPERM=0666

ARTJESADM        SRVGRP =ARTGRP SRVID = 1 MIN=1 MAX=1
   CLOPT = "-A -- -i jesconfig"
   RQPERM=0666 RPPERM=0666

ARTJESCONV       SRVGRP =ARTGRP SRVID = 20 MIN=1 MAX=1
   CLOPT = "-A --"
   RQPERM=0666 RPPERM=0666

ARTJESINITIATOR  SRVGRP =ARTGRP SRVID =30
   CLOPT = "-A -- -n 20 -d"
### Configuring File System Permission

To enable the user substitution successfully, you need to make sure the permission of the file system for users with different identities have been configured to meet the following requirements:

- All job executors should have read and execute permissions to the following folders:
  - $JESDIR/bin/artjesadmin
  - $JESDIR/locale/*
  - $MT_ROOT
- All job executors should have full permissions to the following folders:
  - $DATA_SOURCE
  - $DATA
  - $MT_LOG
  - $SPOOL
  - $MT_TMP
  - $MT_ACC_FILEPATH
  - $JESROOT
- All job users should have read permission to the following folders:
  - $TUXDIR
  - $TUXCONFIG
Note: It is recommended that all job executors have the write permission to ULOG, stdout, and stderr, otherwise the log and output messages cannot be written successfully.

TuxJES Job Operation

TuxJES supports a standard job operation interface. A set of FML32 fields are defined for the service contract. The utility artjesadmin shipped in TuxJES uses typical Tuxedo ATMI APIs (See Oracle Tuxedo API: ATMI for more details) to submit, control, and print job information. Customers are also supported to write personalized applications, which communicate with TuxJES job management system. This documentation explains the general steps and programming approaches.

FML32 Fields in Use for TuxJES Job Operation

All TuxJES job related fields are defined at $JESDIR/udataobj/jesflds. The corresponding header file is at $JESDIR/include/jesflds.h. Table 13 illustrates the fields used in job operation.

Table 13  FML32 Fields in Use for TuxJES Job Operation

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>JES2_JOB_ID</td>
<td>STRING</td>
<td>The job ID generated by TuxJES job management if a job is successfully submitted to the system. The range is [1-99999999].</td>
</tr>
<tr>
<td>JES2_JOB_NAME</td>
<td>STRING</td>
<td>The job name given to this job. It is supplied by users in the shell script.</td>
</tr>
<tr>
<td>JES2_JOB_CLASS</td>
<td>STRING</td>
<td>The job class.</td>
</tr>
<tr>
<td>JES2_JOB_PRTY</td>
<td>SHORT</td>
<td>The job priority ranged in [0-15].</td>
</tr>
<tr>
<td>JES2_JOB_TYPRUNT</td>
<td>STRING</td>
<td>The job type. It is supplied by users for job submission in the shell script.</td>
</tr>
<tr>
<td>JES2_JOBSCRIPT</td>
<td>STRING</td>
<td>The ksh job script name. It can be in the format of absolute path or relative path. The later case will be extended from current working directory.</td>
</tr>
<tr>
<td>JES2_JOB_EJROP</td>
<td>STRING</td>
<td>The options passed to EJR itself.</td>
</tr>
</tbody>
</table>
To use the above fields, you must include header file `$JESDIR/include/jesflds.h` in your source code, and use Tuxedo FML32 APIs to manipulate the FML32 buffer of input and output.

### Table 13  FML32 Fields in Use for TuxJES Job Operation

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>JES2_JOB_OWNER</td>
<td>STRING</td>
<td>The job owner. It is supplied when a job is submitted and returned by server to query job information.</td>
</tr>
<tr>
<td>JES2_JOB_SUBMITTIME</td>
<td>LONG</td>
<td>The submitted timestamp of a job. It is the seconds since Epoch time.</td>
</tr>
<tr>
<td>JES2_JOB_STATUS</td>
<td>STRING</td>
<td>The job status returned by server.</td>
</tr>
<tr>
<td>JES2_JOB_ERROR</td>
<td>STRING</td>
<td>If the operation encounters some errors, the field carries the information.</td>
</tr>
<tr>
<td>JES2_JOB_MSG</td>
<td>STRING</td>
<td>It carries error information for particular jobs. Unlike JES2_JOB_ERROR, JES2_JOB_MSG carries general failure information.</td>
</tr>
<tr>
<td>JES2_JOB_CURRENNT</td>
<td>STRING</td>
<td>The current step of a job.</td>
</tr>
<tr>
<td>JES2_JOB_ENDTIME</td>
<td>LONG</td>
<td>The timestamp of a terminated job. It is seconds since Epoch time.</td>
</tr>
<tr>
<td>JES2_JOB_EXECTIME</td>
<td>LONG</td>
<td>The timestamp of an executing job. It is seconds since Epoch time.</td>
</tr>
<tr>
<td>JES2_JOB_EXECCAC</td>
<td>STRING</td>
<td>The host for the running job.</td>
</tr>
<tr>
<td>JES2_JOB_USRSEC</td>
<td>LONG</td>
<td>The user-level CPU time in seconds.</td>
</tr>
<tr>
<td>JES2_JOBUSRUSEC</td>
<td>LONG</td>
<td>The user-level CPU time for the microseconds portion in one second.</td>
</tr>
<tr>
<td>JES2_JOB_SYSSEC</td>
<td>LONG</td>
<td>The system-level CPU time in seconds.</td>
</tr>
<tr>
<td>JES2_JOB_SYSUSEC</td>
<td>LONG</td>
<td>The system-level CPU time for the microseconds portion in one second.</td>
</tr>
</tbody>
</table>
For more information, please refer to Programming An Oracle Tuxedo ATMI Application Using FML.

**ATMI Services in Use for TuxJES Job Operation**

**ARTJESADM** server provides the following ATMI services as the job handling entries. Client programs can call these services using standard ATMI APIs, such as `tpcall` with FML32 buffer, in which the fields are prepared or retrieved for input and output.

- **JES2SUBMIT**
- **JES2PRINT**
- **JES2HOLD**
- **JES2RELEASE**
- **JES2CANCEL**
- **JES2PURGE**

**JES2SUBMIT**

This service provides the functionality to submit a job.

**Input**

**Required Fields**

- `JES2_JOB_SCRIPT`

**Optional Fields**

- `JES2_JOB_OWNER`

**Note:** If `JES2_JOB_OWNER` is not specified, this job is deemed to "*" user, meaning no ownership of this job.

**Output**

**Success**

Output buffer will contain the following fields.

- `JES2_JOB_ID`
- `JES2_JOB_NAME`
- `JES2_JOB_CLASS`
- `JES2_JOB_TYPRUN`
JES2_JOB_PRTY
- JES2_JOB_OWNER

**Failure**
If the submission fails, JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.

**JES2PRINT**
This service is used to query job information. The interface allows you to get a particular job with job ID, a set of jobs with the same job name and job class, or all jobs. If multiple job instances are got in the results, they will be in FML32 occurrence style, in which occurrence starts at 0. For example, if three jobs are returned, the first job will be at occurrence of 0, the second job will be at 1, and the third job will be at 2.

**Input**

**Required Fields**
None. If nothing specified, the server will print all jobs.

**Optional Fields**
- JES2_JOB_ID (if it is specified, only this job will be retrieved)
- JES2_JOB_NAME (if it is specified, only the jobs of this name will be retrieved)
- JES2_JOB_CLASS (if it is specified, only the jobs of this class will be retrieved)

**Note:** ARTJESADM checks fields in the sequence of job ID, name, and class; once a field is checked, ARTJESADM stops checking.

**Output**

**Success**
If the retrieving is completed successfully, the following fields will be returned.
- JES2_JOB_ID
- JES2_JOB_NAME
- JES2_JOB_CURRENT
- JES2_JOB_STATUS
- JES2_JOB_TYPRUN
- JES2_JOB_PRTY
- JES2_JOB_OWNER
- JES2_JOB_EXECMAC
- JES2_JOB_USRSEC
- JES2_JOB_USRUSEC
- JES2_JOB_SYSSEC
- JES2_JOB_SYSUSEC

Notes:

- JES2_JOB_ID is mandatory to use; others are optional.
- It's suggested to use the occurrence of JES2_JOB_ID as the field manipulation base.
- If there is something wrong with a particular job, JES2_JOB_MSG will be available in the corresponding sequence specified by JES2_JOB_ID. For partially wrong cases, tperrno will be TPOK.

Failure

If a general operation is failed, JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.

JES2HOLD

This service is used to hold a job not in running nor end status. The interface of job collection is the same with JES2PRINT.

Input

Required Fields

None. If nothing specified, the server will hold all jobs.

Optional Fields

- JES2_JOB_ID (if it is specified, only this job will be held)
- JES2_JOB_NAME (if it is specified, only the jobs of this name will be held)
- JES2_JOB_CLASS (if it is specified, only the jobs of this class will be held)

Note: ARTJESADM checks fields in the sequence of job ID, name, and class; once a field is checked, ARTJESADM stops checking.
Output

Success
If the operation is completed successfully, the following fields will be returned.

- JES2_JOB_ID
- JES2_JOB_NAME

Note: If there is something wrong with a particular job, JES2_JOB_MSG will be available in the corresponding sequence specified by JES2_JOB_ID. For partially wrong cases, tperrno will be TPOK.

Failure
If a general operation is failed, the JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.

JES2RELEASE
This service is used to release a job in hold status. The job can be in hold status by a hold operation, submitting time typerun. The interface of job collection is the same with JES2PRINT.

Input

Required Fields
None. If nothing specified, the server will release all jobs.

Optional Fields

- JES2_JOB_ID (if it is specified, only this job will be released)
- JES2_JOB_NAME (if it is specified, only the jobs of this name will be released)
- JES2_JOB_CLASS (if it is specified, only the jobs of this class will be released)

Note: ARTJESADM checks fields in the sequence of job ID, name, and class; once a field is checked, ARTJESADM stops checking.

Output

Success
If the operation is completed successfully, the following fields will be returned.

- JES2_JOB_ID
- JES2_JOB_NAME
Note: If there is something wrong with a particular job, JES2_JOB_MSG will be available in the corresponding sequence specified by JES2_JOB_ID. For partially wrong cases, tperrno will be TPOK.

Failure
If a general operation is failed, JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.

JES2CANCEL
This service is used to cancel a job. The interface of job collection is the same with JES2PRINT.

Input
Required Fields
None. If nothing specified, the server will cancel all jobs.

Optional Fields
- JES2_JOB_ID (if it is specified, only this job will be canceled)
- JES2_JOB_NAME (if it is specified, only the jobs of this name will be canceled)
- JES2_JOB_CLASS (if it is specified, only the jobs of this class will be canceled)

Note: ARTJESADM checks fields in the sequence of job ID, name, and class; once a field is checked, ARTJESADM stops checking.

Output
Success
If the operation is completed successfully, the following fields will be returned.
- JES2_JOB_ID
- JES2_JOB_NAME

Note: If there is something wrong with a particular job, JES2_JOB_MSG will be available in the corresponding sequence specified by JES2_JOB_ID. For partially wrong cases, tperrno will be TPOK.

Failure
If a general operation failed, JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.
**JES2PURGE**

This service is used to purge a job. If a job is not finished yet, the effect is the same with JES2CANCEL. The interface of job collection is the same with JES2PRINT.

**Input**

**Required Fields**

None. If nothing specified, the server will purge all jobs.

**Optional Fields**

- JES2_JOB_ID (if it is specified, only this job will be purged)
- JES2_JOB_NAME (if it is specified, only the jobs of this name will be purged)
- JES2_JOB_CLASS (if it is specified, only the jobs of this class will be purged)

**Note:** ARTJESADM checks fields in the sequence of job ID, name, and class; once a field is checked, ARTJESADM stops checking.

**Output**

**Success**

If the operation is completed successfully, the following fields will be returned.

- JES2_JOB_ID
- JES2_JOB_NAME

**Note:** If there is something wrong with a particular job, JES2_JOB_MSG will be available in the corresponding sequence specified by JES2_JOB_ID. For partially wrong cases, tperrno will be TPOK.

**Failure**

If a general operation is failed, JES2_JOB_ERROR carries the error message and tperrno is set to TPESVCFAIL.

**Event Subscriptions for Job Status Changes**

TuxJES provides an event notification mechanism based on Tuxedo EventBroker (See Subscribing to Events for more details). The following user-level events are provided when something happens at each stage. The data buffer of each event is FML32 buffer for a particular job.
Since the event post depends on JESCONFIG configurations in ARTJESADM, the switch must be turned on if all or some event want to.

### Security Consideration

The application using the job management service interface must be compliant to Tuxedo ATMI security convention (See Introducing ATMI Security for more details). The proper application password and username/password must be prepared for `tpinit()` in a client application. To control the permission of services, adequate ACL or MANDATORY ACL should be configured in UBBCONFIG for the services listed in ATMI Services in Use for TuxJES Job Operation.

### Sample of Submitting a Job

The following sample demonstrates how to submit a job. `-I $JESDIR/include` must include `CFLAGS`.

---

**Table 14 Available TuxJES Events for Subscription**

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARTJES_JOBSUBMIT</td>
<td>A new job is submitted.</td>
</tr>
<tr>
<td>ARTJES_JOBCVT</td>
<td>A job conversion is completed.</td>
</tr>
<tr>
<td>ARTJES_JOBCVT_FAIL</td>
<td>A job conversion fails.</td>
</tr>
<tr>
<td>ARTJES_JOBEXEC</td>
<td>A job is finished successfully.</td>
</tr>
<tr>
<td>ARTJES_JOBEXEC_FAIL</td>
<td>A job's execution fails.</td>
</tr>
<tr>
<td>ARTJES_JOBPURGE</td>
<td>A job is purged.</td>
</tr>
<tr>
<td>ARTJES_JOBCANCEL</td>
<td>A job is canceled.</td>
</tr>
</tbody>
</table>
Listing 9  Sample of submitting a job

```c
#include <stdio.h>
#include <tpadm.h>
#include <fml32.h>
#include <jesflds.h>

main(int argc, char *argv[])
{
    FBFR32 *input;
    FBFR32 *output;
    long buflen;

    if (tpinit((TPINIT *) NULL) == -1) {
        (void)fprintf(stderr, "Tpinit failed: %s\n", tpstrerror(tperrno));
        exit(1);
    }

    input = tpalloc("FML32", NULL, 1024);
    output = tpalloc("FML32", NULL, 1024);

    if (input == NULL || output == NULL) {
        (void)fprintf(stderr, "Tpinit failed: %s\n", tpstrerror(tperrno));
        exit(1);
    }

    if (Fappend32(input, JES2_JOB_SCRIPT, "JOBA", 0) < 0 ) {
        (void)fprintf(stderr, " Fappend32 failed: \n");
        exit(1);
    }
}
```
Sample of Submitting a Job


)
Findex32(input, 0);
if (tpcall("JES2SUBMIT", input, 0, &output, &buflen, 0) < 0 ) {
    (void) fprintf(stderr, "Can't send request to service JES2SUBMIT");
    exit(1);
} else {
    char *jobid;
    jobid = Ffind32(output, JES2_JOB_ID, 0, NULL);
    if (jobid != NULL) {
        fprintf(stdout, "Job %s is submitted\n", jobid);
    }
}
}


return(0);
}
Tuxedo Job Enqueueing Service (TuxJES)