

# Oracle Tuxedo Application Runtime for Batch

Reference Guide

12c Release 2 (12.1.3)

April 2014

ORACLE®

Copyright © 2010, 2014 Oracle and/or its affiliates. All rights reserved.

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

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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

# 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

# Introduction

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

# Z/OS JCL in the Batch Runtime Environment

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**

JCL Card	Parameter	Equivalent in Target Environment	Status
//*		Comments	supported
/*		In-stream data delimiter	supported
//		End of job	supported
COMMAND			N.S.
CNTL/ENDCNTL			N.R.



**Table 2-1 JCL Card Equivalences**

JCL Card	Parameter	Equivalent in Target Environment	Status
DD	*	m_FileAssign -i	supported
	ACCODE		N.R.
	AMP		N.R.
	AVGREC		N.R.
	BLKSIZE		N.R.
	BLKSZLIM		N.R.
	CCSID		N.R.
	CHKPT		N.S.
	CNTL		N.R.
	DATA	m_FileAssign	supported
	DATACLAS	see <a href="#">DATACLAS Parameter of DD Statement</a>	supported
	DCB	see <a href="#">DCB Parameter of DD Statement</a>	supported
	DISP	m_FileAssign -d <DISP option>	supported
	DLM	m_FileAssign -D <delimiter>	supported
	DSID		N.R.
DSNAME	m_FileAssign	supported	

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
DD (continued)	DSNTYPE		N.S.
	DUMMY	m_FileAssign with /dev/null	supported
	DYNAM	m_FileAssign with /dev/null	supported
	EXPDT	m_FileAssign -e	supported
	FILEDATA		N.R.
	FREE		N.R.
	KEYLEN	m_FileAssign -k	supported
	KEYOFF	m_FileAssign -k	supported
	LABEL		N.R.
	LGSTREAM		N.R.
	LIKE	m_FileAssign -s	supported
	LRECL	m_FileAssign -r	supported
	MGMTCLAS		N.R.
	MSVGP		N.R.
	PATH		N.R.
	PATHDISP		N.R.
	PATHMODE		N.R.
	PATHOPTS		N.R.
	PROTECT		N.R.
	QNAME		N.R.
RECFM	m_FileAssign -r	supported	
RECORD	m_FileAssign -t	supported	

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
DD (continued)	REFDD		N.S.
	RETPD	m_FileAssign -e	supported
	RLS		N.R.
	SECMODEL		N.R.
	SPACE		N.R.
	STORCLAS		N.R.
	SUBSYS		N.S.
	TERM		N.R.
	UNIT		N.R.
	VOLUME	m_FileAssgin -v	supported

Table 2-1 JCL Card Equivalences

JCL Card	Parameter	Equivalent in Target Environment	Status
DD : Printing parameters	BURST		N.S.
	CCSID		N.R.
	CHARS		N.S.
	COPIES	<b>m_OutputAssign -c</b>	supported
	DEST	<b>m_OutputAssign -d</b>	supported
	FCB		N.S.
	FLASH		N.S.
	HOLD	<b>m_OutputAssign -H</b>	supported
	MODIFY		N.S.
	OUTLIM		N.S.
	OUTPUT	<b>m_OutputAssign -o</b>	supported
	SEGMENT		N.R.
	SPIN		N.S.
	SYSOUT	<b>see SYSOUT Parameter of DD Statement</b>	supported
UCS		N.S.	

Table 2-1 JCL Card Equivalences

JCL Card	Parameter	Equivalent in Target Environment	Status
DD: Special statements	JOBLIB	<b>m_JobLibset</b>	supported
	STEPLIB	<b>m_StepLibset</b>	supported
	SYSABEND		N.S.
	SYSMDUMP		N.S.
	SYSUDUMP		N.S.
	SYSCHK		N.S.
	SYSIN	<b>m_FileAssign</b>	supported
	SYSCKEOV		N.S.
EXEC	ACCT		N.S.
	ADDRSPC		N.R.
	CCSID		N.R.
	COND	<b>m_CondExec</b>	supported
	DPRTY		N.R.
	DYNAMNBR		N.R.
	MEMLIMIT		N.R.
	PARM	<b>m_ProgramExec</b>	supported
	PERFORM		N.R.
	PGM	<b>m_ProgramExec</b>	supported
	PROC		supported
	RD		N.S.
	REGION		N.R.
	RLSTMOUT		N.R.
	TIME		N.S.

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
IF THEN ELSE END		<b>m_CondIf</b>  <b>m_CondElse</b> <b>m_CondEndif</b>	supported
INCLUDE	MEMBER	<b>m_ShellInclude</b>	
JCLLIB	ORDER	<b>m_JclLibSet</b>	

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
JOB	<jobname>	m_JobBegin	supported
	<accounting>		N.S.
	<prog.name>		N.S.
	ADDRSPC		N.R.
	BYTES		N.S.
	CARDS		N.R.
	CCSID		N.S.
	CLASS	m_JobBegin -c (with TuxJES).	supported
	COND	m_JobBegin -C	supported
	GROUP		N.R.
	JESLOG		N.S.
	LINES		N.S.
	MEMLIMIT		N.R.
	MSGCLASS		N.S.
	MSGLEVEL		N.S.
	NOTIFY		N.S.
	PAGES		N.S.
	PASSWORD		N.R.
PERFORM		N.R.	
PRTY	m_JobBegin -p (with TuxJES).	supported	

**Table 2-1 JCL Card Equivalences**

JCL Card	Parameter	Equivalent in Target Environment	Status
JOB	RD		N.S.
	REGION		N.R.
	RESTART	see <a href="#">RESTART Parameter of JOB Statement</a>	Partially supported
	SECLABEL		N.R.
	SCHENV		N.R.
	TIME		N.S.
	TYPRUN	m_JobBegin -t (with TuxJES)	supported
	USER		N.S.



**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
OUTPUT	ADDRESS		N.S.
	AFPSTATS		N.S.
	BUILDING		N.S.
	BURST		N.S.
	CHARS		N.S.
	CKPTLINE		N.S.
	CKPTPAGE		N.S.
	CKPTSEC		N.S.
	CLASS	m_OutputSet -c	supported
	COLORMAP		N.S.
	COMPACT		N.S.
	COMSETUP		N.S.
	CONTROL		N.S.
	COPIES	m_OutputSet -n	supported
	DATAACK		N.S.
	DEFAULT	m_OutputSet -D	supported
	DEPT		N.S.
	DEST	m_OutputSet -d	supported

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
OUTPUT (continued)	DPAGELBL		N.S.
	DUPLEX		N.S.
	FCB		N.S.
	FLASH		N.S.
	FORMDEF		N.S.
	FORMLEN		N.S.
	FORMS	m_OutputSet -f	supported
	FSSDATA		N.S.
	GROUPID		N.S.
	INDEX		N.S.
	INTRAY		N.S.
	JESDS		N.S.
	LINDEX		N.S.
	LINECT		N.S.
	MAILBCC		N.S.
	MAILCC		N.S.
	MAILFILE		N.S.
	MAILFROM		N.S.
	MAILTO		N.S.
	MODIFY		N.S.
NAME		N.S.	
NOTIFY		N.S.	
OFFSETXB		N.S.	

**Table 2-1 JCL Card Equivalences**

<b>JCL Card</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
OUTPUT (continued)	OFFSETXF		N.S.
	OFFSETYB		N.S.
	OFFSETYF		N.S.
	OUTBIN		N.S.
	OUTDISP		N.S.
	OVERLAYB		N.S.
	OVERLAYF		N.S.
	OVFL		N.S.
	PAGEDEF		N.S.
	PIMSG		N.S.
	PORTNO		N.S.
	PRMODE		N.S.
	PRTATTRS		N.S.
	PRTEROR		N.S.
	PRTOPTNS		N.S.
	PRTQUEUE		N.S.
	PRTY	m_OutputSet -p	supported
	REPLYTO		N.S.
	RESFMT		N.S.
	RETAINF		N.S.
RETAINS		N.S.	
RETRYL		N.S.	
RETRYT		N.S.	

**Table 2-1 JCL Card Equivalences**

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

## DCB Parameter of DD Statement

**Table 2-2 DCB Parameter of DD Statement**

<b>DCB parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
BFANL		N.R.
BFTEK		N.R.
BLKSIZE		N.R.
BUFIN		N.R.
BUFL		N.R.
BUFMAX		N.R.
BUFNO		N.R.
BUFOFF		N.R.
BUFOUT		N.R.
BUFSIZE		N.R.
CPRI		N.R.
CYLOFL		N.R.
DEN		N.R.
DIAGNS		N.R.
DSORG	<code>m_FileAssign -t</code>	supported
EROPT		N.R.
FUNC		N.R.
GNCPL		N.R.
INTVL		N.R.
IPLTXID		N.R.
KEYLEN	<code>m_FileAssign -k</code>	supported

**Table 2-2 DCB Parameter of DD Statement**

<b>DCB parameter</b>	<b>Equivalent in Target Environment</b>	<b>Status</b>
LIMCT		N.R.
LRECL	m_FileAssign -r	supported
MODE		N.R.
NCP		N.R.
NTM		N.R.
OPTCD		N.R.
PCI		N.R.
PRTSP		N.R.
RECFM	m_FileAssign -r	supported
RESERVE		N.R.
RKP	m_FileAssign -t	supported
STACK		N.R.
TRESH		N.R.
TRTCH		N.R.

## SYSOUT Parameter of DD Statement

**Table 2-3 SYSOUT Parameter of DD Statement**

SYSOUT parameter	Usage	Equivalent in Target Environment	Status
First parameter	Class	m_OutputAssign -c	supported
Second parameter	INTRDR	m_OutputAssign -w	supported
	SMTP	m_OutputAssign -w	supported
	<writer>	m_OutputAssign -w	supported
Third parameter	Form name	m_OutputAssign -f	supported
	Code name		N.S.

## 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 2-4 Supported Parameters**

Statement	Supported Parameter
<code>/*XEQ</code>	<code>/*XEQ nodename</code>
<code>/*ROUTE</code>	<code>/*ROUTE PRINT LOCAL</code> <code>/*ROUTE PRINT nodename</code> <code>/*ROUTE PUNCH nodename</code> <code>/*ROUTE XEQ nodename</code>
<code>/*XMIT</code>	<code>/*XMIT nodename</code>

## ABEND Routine

Captures terminations abnormally or normally from user program. It provides three COBOL programs, `ILBOABN0`, `CEE3ABD` and `ART3ABD`, which can be called in specific conditions to `ABEND` the program by itself. When the `ABEND` routine calls `stop run`, `stop run` terminates the process after Batch Runtime performs necessary cleanup.



- ILBOABN0 abcode

abcode: the ABEND code.

- CEE3ABD abcode, clean-up

abcode: the ABEND code.

- clean-up(input)

The possible values are:

0: Issue the ABEND. Batch Runtime will capture ABEND and do clean-up work. Job failed.

1: Issue the ABEND. Batch Runtime will capture ABEND and store a system dump in a file, then do clean-up work. Job failed.

- ART3ABD abcode

abcode: the ABEND code.

The ILBOABN0.cbl source is in the Batch\_RT/ejr/SAMPLE directory.

**Note:** For Micro Focus COBOL, Batch Runtime provides two ABEND-handling routines, ILBOABN0 and ART3ABD, that are called to force ABEND. Micro Focus COBOL provides its own ABEND-handling routine, CEE3ABD. It works as defined by Micro Focus COBOL.

For COBOL-IT COBOL, Batch Runtime provides two ABEND routines, ILBOABN0 and CEE3ABD.

## General Utility Commands Equivalence Table

Table 2-5 General Utility Commands Equivalences

Utility	Command	Equivalent in Target Environment	Status
IDCAMS (modal commands)	IF/THEN/ELSE		supported
	SET		supported
	CANCEL		N.S.
	DO/END		N.S.
	MAXCC		supported
	LASTCC		supported

**Table 2-5 General Utility Commands Equivalences**

Utility	Command	Equivalent in Target Environment	Status
IDCAMS	ALLOCATE (main parameters)	m_FileBuild	supported
	ALTER NEWNAME	m_FileRename	supported <b>Note:</b> Renaming for VSAM RDB files is not supported.
	ALTER LIBRARYENTRY		N.R.
	ALTER VOLUMEENTRY		N.R.
	BLDINDEX		N.R.
	CREATE LIBRARYENTRY		N.R.
	CREATE VOLUMEENTRY		N.R.
	DCOLLECT		N.S.
	DEFINE ALIAS		N.S.
	DEFINE AIX	m_FileBuild -K	supported
	DEFINE CLUSTER	m_FileBuild	supported
	DEFINE GDG	m_GenDefine	supported
	DEFINE NONVSAM		N.S.
	DEFINE PAGESPACE		N.R.
	DEFINE PATH		N.R.
	DEFINE USERCATALOG		N.R.
	DELETE ALAIS		N.S.
	DELETE AIX	m_FileBuild -R	supported
	DELETE CLUSTER	m_FileDelete	supported
DELETE GDG	m_FileDelete	supported	

**Table 2-5 General Utility Commands Equivalences**

Utility	Command	Equivalent in Target Environment	Status
IDCAMS (continued)	DELETE NVR		N.S.
	DELETE PAGESPACE		N.R.
	DELETE PATH		N.R.
	DELETE TRUENAME		N.S.
	DEL USERCATALOG		N.R.
	DELETE VOLUMEENTRY		N.R.
	DELETE VVR		N.R.
	DIAGNOSE		N.S.
	EXAMINE		N.S.
	EXPORT		N.S.
	EXPORT DISCONNECT		N.R.
	IMPORT		N.S.
	IMPORT CONNECT		N.R.
	LISTCAT	m_FileListcat	supported
	LISTDATA		N.S.
	PRINT	m_FilePrint	supported
	REPRO	m_FileRepro	supported
	SETCACHE		N.R.
	SHCDS		N.R.
	VERIFY		N.S.

**Table 2-5 General Utility Commands Equivalences**

Utility	Command	Equivalent in Target Environment	Status
IEBCOPY	ALTERMOD		N.R.
	COPY	m_DirCopy	supported
	COPYGRP		N.R.
	COPYMOD		N.R.
	EXCLUDE	m_DirCopy -e	supported
	SELECT	m_DirCopy -s	supported
IEBGENER	With SYSIN DUMMY	m_FileLoad	supported
	GENERATE MEMBER		N.S.
	GENERATE RECORD	m_FileSort	supported
IEFBR14		m_ProgramExec IEFBR14	supported
IEFBR15		m_ProgramExec IEFBR15	supported
PKZIP	Main parameters	m_Pkzip	supported
PKUNZIP	Main parameters	m_Pkunzip	supported
SMTP		m_Smtp	supported
XMITIP		m_SendMail	supported
ZIP390	ACTION : ZIP	m_Pkzip	supported
	ACTION : UNZIP	m_Pkunzip	supported
	ENCRYPT		N.S.
	IFILE		supported
	OFFILE		supported
	ARCHIVE		supported
	OVERWRITE		supported

# Sort Utilities

## Sort Utilities Equivalence Tables

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

**Table 2-6 Sort Utilities Equivalences**

<b>Sort Statement</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>MFSORT</b>	<b>SyncSort</b>
SORT		m_FileSort	Supported	Supported
MERGE		m_FileSort	Supported	Supported
OPTION		m_FileSort		
	COPY		Supported	Supported
	SKIPREC		Supported	Supported
	STOPAFT		Supported	Supported
	INCLUDE		Supported	Supported
	OMIT		Supported	Supported
	OUTFILE		Supported	Supported
	OUTREC		Supported	Supported
	END		Supported	Supported
	RECORD		Supported	Supported
	SUM		Supported	Supported

## ICETOOL

**Table 2-7 ICETOOL Equivalences**

Sort Statement	Equivalent in Target Environment	MFSORT	SyncSort
SORT	m_FileSort	supported	supported
COPY	m_FileSort	supported	supported
MERGE	m_FileSort	supported	supported

## 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**

Command	Parameter	Equivalent in Target Environment	Oracle	DB2LUW
DSNTEP2	SQL	m_ExecSQL	supported	supported
DSNTEP4	SQL	m_ExecSQL	supported	supported
DSNTIAD	SQL	m_ExecSQL	supported	supported
DSNTIAUL	SQL	m_ExecSQL	supported	supported
	"Unload"	m_DBTableUnload	supported	supported

**Table 2-8 DB2 Utilities**

<b>Command</b>	<b>Parameter</b>	<b>Equivalent in Target Environment</b>	<b>Oracle</b>	<b>DB2LUW</b>
DSNUTILB	EXEC SQL	m_ExecSQL	supported	supported
	LOAD	m_DBTableLoad	supported	supported
	UNLOAD	m_DBTableUnload	supported	supported

## Z/OS JCL in the Batch Runtime Environment



# Oracle Tuxedo Application Runtime for Batch Functions

This chapter contains the following topics:

- [Introduction to the Batch Runtime Commands](#)
- [EJR Syntax](#)
- [Oracle Tuxedo Application Runtime for Batch Functions](#)
- [Reference](#)

## 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

```
EJR [-v] [-s sysoutDir] [-L LogDir] [-H] [-R StepName] [-d regexp] [-f
EnvFile] [-t file] [-V n] [-l] [-J] Job
```

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, where each item should follow this format:

```
MY_ENV_NAME=MY_ENV_VALUE; export MY_ENV_NAME
```

**-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 `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 `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 `.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 EJER, a log file is generated. When not using TuxJES, the name of the log file is:

`JobName_YYYYMMDDHHMMSS_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 3-1 Oracle Tuxedo Application Runtime for Batch Return Codes**

Return Code	Description
<code>MT_RC</code>	<p>The Return code for an Oracle Tuxedo Application Runtime for Batch function execution.</p> <ul style="list-style-type: none"> <li>• if <code>MT_RC</code> = Cnnnn, return code OK</li> <li>• if <code>MT_RC</code> = Unnnn, User abort</li> <li>• if <code>MT_RC</code>= Syyy, System abort</li> </ul>
<code>MT_RC_JOB</code>	<p>General return code (for the job) <code>MT_RC_JOB</code> is updated with <code>MT_RC</code> at the end of each phase. It contains the maximum <code>MT_RC</code> value for the job.</p>
<code>MT_RC_ABORT</code>	<p>Value fixed to D0000.</p> <p>The <code>MT_RC</code> return code is compared to this value to determine if the result of the execution was normal or not.</p>

**Table 3-1 Oracle Tuxedo Application Runtime for Batch Return Codes**

<b>Return Code</b>	<b>Description</b>
MT_RC_PROGRAM_ABORT	<p>By default 127.</p> <p>Contains the return code from an executable (not applicative COBOL program).</p> <ul style="list-style-type: none"> <li>● from 0 to 127: return code OK</li> <li>● 128: Unnn in MT_RC</li> <li>● from 129 to 142: it is a signal SKILL in MT_RC</li> <li>● more than 143: Unnn in MT_RC</li> </ul>
MT_RC_STEP_RETURNCODE_{LABEL}	<p>Each phase return code is saved. The variable name contains the phase label. This variable can be used for specific chaining within the script.</p>

**Table 3-2 EJR Return Code**

<b>Return Code</b>	<b>Description</b>
0	<p>Normal end</p> <p>Normal execution with user error</p>
1	System error (abort)
5	No present of JOBID on command line
6	No present of JES2ROOT on command line
11	Some files or directories used for JES2 are not found
12	Access right problem
13	Write error - File is not created or written
21	The KSH script is not found
22	The KSH script is not readable
23	The KSH script is not executable

**Table 3-2 EJR Return Code**

<b>Return Code</b>	<b>Description</b>
50	Cancel by user
51	Cancel by system
87	PID is not running (killing process)
52 <= RC <= 99	Temporary Runtime error (the error may appear in case of restart) For example: no enough space in technical temporary storage
123	Default return code must be changed during treatment
125	Bad verbose level (in case of using option v)
> =100	Permanent Runtime error (the error appears in case of restart) For example: an missing of batch RT component

## 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 optarg 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**

Functions		
<code>m_CondElse</code>	<code>m_FileListcat</code>	<code>m_OutputOverride</code>
<code>m_CondEndif</code>	<code>m_FileLoad</code>	<code>m_OutputSet</code>
<code>m_CondExec</code>	<code>m_FileOverride</code>	<code>m_PhaseBegin</code>
<code>m_CondIf</code>	<code>m_FilePrint</code>	<code>m_PhaseEnd</code>
<code>m_DBTableLoad</code>	<code>m_FileRepro</code>	<code>m_ProcBegin</code>
<code>m_DirCreate</code>	<code>m_FileSort</code>	<code>m_ProcEnd</code>
<code>m_DirDelete</code>	<code>m_FileRename</code>	<code>m_ProcInclude</code>
<code>m_DirRename</code>	<code>m_FileLoad</code>	<code>m_ProgramExec</code>
<code>m_DSNUTILB</code>	<code>m_FileOverride</code>	<code>m_RcSet</code>
<code>m_ExecSQL</code>	<code>m_GenDefine</code>	<code>m_ShellInclude</code>
<code>m_ExpdtAlter</code>	<code>m_IDCAMS</code>	<code>m_StepLibSet</code>
<code>m_FileAssign</code>	<code>m_IDCAMSSetCC</code>	<code>m_SymbolDefault</code>
<code>m_FileBuild</code>	<code>m_JclLibSet</code>	<code>m_SymbolSet</code>
<code>m_FileClrData</code>	<code>m_JobBegin</code>	<code>m_UtilityExec</code>
<code>m_FileDelete</code>	<code>m_JobEnd</code>	<code>m_SetJobExecLocation</code>
<code>m_FileEmpty</code>	<code>m_JobLibSet</code>	<code>m_PhaseEnd</code>
<code>m_FileExist</code>	<code>m_OutputAssign</code>	<code>m_ProcBegin</code>

## Overview

The functions correspond to the interface (API) between the shell script and the Batch Runtime executable. Some scripts, such as `m_JclibSet`, 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] [-a] [-r]
```

### Description

When `MT_DSNUTILB_LOADUNLOAD` is set to "yes", this function calls the COBOL program "schema-table-L" for loading data from a file into table in DB, and `MT_CTL` is not used. The COBOL program is generated by Workbench Rdbms Convertor.

When `MT_DSNUTILB_LOADUNLOAD` is set to other value than "yes", This function executes a command stored in the file which dname 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. When `MT_DSNUTILB_LOADUNLOAD` is set to "yes", the value for this option can be one of the following two formats: `TableName`, `SchemaName.TableName`

When the value does not contain schema name, the called COBOL program name is:  
`$(MT_DB_DEFAULT_SCHEMA)-TableName-L`

When the value contains schema name, the called COBOL program name is:  
`SchemaName-TableName-L`

**-i <input flat file>**

Mandatory option.

The ddname of the file containing the data to be loaded. When

MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option is ignored.

## 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.

When MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option is ignored.

**-d <discard file>**

Optional.

SYSERR by default.

The ddname of the file that contains the discarded data during sqlldr command execution.

When MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option is ignored.

**-D <discard file>**

Optional.

999 by default

The maximum number of discarded data. When MT\_DSNUTILB\_LOADUNLOAD is set to

"yes", this option is ignored.

**-a**

Optional.

When this option is set, the data to be loaded is appended to the end of the table; otherwise, this function returns "8" when table is not empty.

Only when MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option takes effect.

**-r**

Optional.

When this option is set, the table is cleaned before loading data into table; otherwise, the table is not cleaned before loading data into table.

Only when MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option takes effect.



## Options for DB2LUW

### **-e <error file>**

Optional.

<Logfile> by default.

The ddname of the log file used during db2 load command execution. When MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this option is ignored.

### **-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_DBTableLoad -t SCHEMA.TABLE -I DDIN -a -r
```

## m\_DBTableUnload

### Name

m\_DBTableUnload – Unloads the content of a database table into a flat file.

### Synopsis

```
m_DBTableUnload -t -o
```

### Description

When MT\_DSNUTILB\_LOADUNLOAD is set to "yes", this function calls the COBOL program "schema-table-U" for loading data from a file into table in DB, and MT\_CTL is not used. The COBOL program is generated by Workbench Rdbms Converter.

When MT\_DSNUTILB\_LOADUNLOAD is set to other value than "yes", 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

### **-t <table name>**

Mandatory option.

The name of the database table to be unloaded.

When `MT_DSNUTILB_LOADUNLOAD` is set to "yes", the value for this option can be one of the following two formats: `TableName`, `SchemaName.TableName`

When the value doesn't contain schema name, the called COBOL program name is:

`${MT_DB_DEFAULT_SCHEMA}-TableName-U`

When the value contains schema name, the called COBOL program name is:

`SchemaName-TableName-U`

### **-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_DBTableUnload -t SCHEMA.TABLE -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.

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

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

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

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

```
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\_DSNUTILB****Name**

m\_DSNUTILB – Mainframe simulation utility that loads/unloads table from database.

**Synopsis**

```
m_DSNUTILB
```

**Description**

m\_DSNUTILB uses DD SYSIN as input. DD SYSIN includes standard DSNUTILB utility commands. Currently, only load and unload commands are supported.

## Example(s)

### Listing 3-1

---

```
(START)

    m_OutputAssign -c A    SYSOUT
    m_OutputAssign -c "*"  SYSPRINT
    m_FileAssign -i SRCIN

UNLOAD UNLDDN OUTFILEA FROM TABLE ZS

LOAD  DATA INDDN OUTFILEB RESUME YES INTO TABLE ZS

UNLOAD UNLDDN OUTFILEC FROM TABLE ARTBATCH.ZS

LOAD  DATA INDDN OUTFILED REPLACE YES INTO TABLE ARTBATCH.ZS

UNLOAD UNLDDN OUTFILEE FROM TABLE ARTBATCH.ZS

_end

    m_FileAssign -d NEW,KEEP,KEEP OUTFILE ${DATA}/dynsysin
    m_FileRepro -i SRCIN -o OUTFILE

    JUMP_LABEL=UNLOAD_1

    ;;

(UNLOAD_1)

    m_FileAssign -d NEW,KEEP,DELETE OUTFILEA
${DATA}/TABLE_ZS_UNLOAD_1.txt

    m_FileAssign -d OLD,KEEP,KEEP OUTFILEB
${DATA}/TABLE_ZS_UNLOAD_1.txt

    m_FileAssign -d NEW,KEEP,DELETE OUTFILEC
${DATA}/TABLE_ZS_UNLOAD_2.txt

    m_FileAssign -d OLD,KEEP,KEEP OUTFILED
${DATA}/TABLE_ZS_UNLOAD_1.txt

    m_FileAssign -d NEW,KEEP,DELETE OUTFILEE
${DATA}/TABLE_ZS_UNLOAD_3.txt
```

```

m_FileAssign -d OLD,KEEP,KEEP SYSIN ${DATA}/dynsysin
m_DSNUTILB
JUMP_LABEL=CLEAN

; ;

```

---

## 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;
_end
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 data base 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.

yyyyddd

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

Note: Expiration dates of [19]99365 and [19]99366 are considered “never-scratch” dates.

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 DSNAMES are 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 DSNAMES 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 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.

Note: Expiration dates of [19]99365 and [19]99366 are considered “never-scratch” dates.

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:

```
m_FileAssign -d SHR ENTREE ${DATA}/PJ01DDD.BT.QSAM.KBIEI001
```

Example with a sysin and a delimiter:

```
m_FileAssign -i -D FF INPUT
```

```
data input 1
```

```
data input 2
```

```
FF
```

Example with a sysin and continuation:

```
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:

```
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

```
m_FileBuild [-t][-r][-k][-K] [-S]<filename> [-v volume] [-e Ydate|Ddays]
dsname
```

### 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 and SEQ 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

yyyyddd

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

Note: Expiration dates of [19]99365 and [19]99366 are considered “never-scratch” dates.

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}/METAW00.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}/METAW.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.KBSTO045
```

## 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]
[-r] [-e]
```

**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.

**-r**

Optional: It specifies whether to replace records in output file by the records with the same key, if any, in input file. This option is applicable to only KSDS file. Considering that there is a record in output file (called old record) and a record with the same key in input file (called new record), if `-r` is specified, the old record is replaced with the new record; otherwise, the old record is kept in output file.

**-e**

Optional: Empties output file before adding input file records. This option is applicable only to KSDS files.

## Example(s)

```
m_FileRepro -I ${DATA}/INPUT -C 1
m_FileRepro -i INPUT -C 5
m_FileRepro -I ${DATA}/INPUT -O ${DATA}/OUTPUTKSDS -r
m_FileRepro -I ${DATA}/INPUT -O ${DATA}/OUTPUTKSDS -e
```

## 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

```
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. It launches an ftp process and executes ftp command(s) stored in an input file. An environment variable, `S` must be declared and initialized with the following value (upper or lower case):

`Y` or `YES`: Test mode. Ftp commands (except for “open”, “user”, “quit” or “bye”), are not executed.

`N` or `NO`: real mode. All ftp commands are executed.

\\$MT\_FTP\_PASS can be added to the input file to replace the real ftp password (which means the security profile password is used). For example: user user1 \\$MT\_FTP\_PASS.

## 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.

Note: Expiration dates of [19]99365 and [19]99366 are considered “never-scratch” dates.

**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\_IDCAMS****Name**

m\_IDCAMS — Executes the stored IDCAMS commands.

## Synopsis

```
m_IDCAMS [sysin file]
```

## Description

Executes the contents of an external sysin. This function executes the script in the current Shell. The script assigned with sysin should contain a sequence of IDCAMS valid RunTime Batch functions.

## Options

**sysin file:** By default `SYTSIN`. ddname of a `sysin` file which contains function to launch.

## Example(s)

```
m_FileAssign -d NEW SYTSIN ${SYSIN}/SYSTIN/IDCAMS.sysin
m_IDCAMS SYTSIN
```

The file `IDCAMS.sysin` may have the following content:

### Listing 3-2 IDCAMS.sysin

---

```
m_FileDelete ${DATA}/INFP.INF.RPG0002.INFP3D
m_CondIf "MAXCC,EQ,0"
m_FileRepro -I ${DATA}/INFP.INF.RPG0001 -O ${DATA}/INFP.INF.RPG0002.INFP3D
m_CondElse
m_FileRepro -I ${DATA}/INFP.INF.RPG0002 -O ${DATA}/INFP.INF.RPG0002.INFP3D
m_CondEndif
```

---

## m\_IDCAMSSetCC

### Name

`m_IDCAMSSetCC` — Sets IDCAMS condition code.

### Synopsis

```
m_IDCAMSSetCC Condition ConditionCode
```



## Description

m\_IDCAMSSetCC sets the IDCAMS condition code

## Options

Condition: Value can be LASTCC or MAXCC

ConditionCode: The new value of the condition code to be set.

## Example(s)

```
m_IDCAMSSetCC LASTCC 16
m_IDCAMSSetCC MAXCC 8
m_IDCAMSSetCC MAXCC 0
```

## m\_JclLibSet

### Name

m\_JclLibSet — Specify conversion stage Procedure and Include directories.

### Synopsis

```
m_JclLibSet directory
```

### Description

m\_JclibSet 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 Micro Focus 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 RDRICICO must contain a ksh script.

```
m_OutputAssign -c R -w INTRDR RDRICICO
```

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" "-F" and "-n".

```
-f `ddn1,ddn2,ddn3,pds;member1;member2`
-F `dsn1,dsn2,dsn3`
-n `,newddn2,newddn3,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 into which 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 [BatchRT.conf](#) file):

- MT\_EXCI
- MT\_JESDECRYPT
- MT\_EXCI\_XA
- MT\_EXCIGRPNAME

MT\_EXCI is set to a 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 a 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 MT\_UTILITY\_LIST\_UN SUPPORT.

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 MT\_UTILITY\_LIST\_UN SUPPORT 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 BPRAB006 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

```
m_RcSet ${MT_RC_ABORT:-S999} "Unknown label : ${CURRENT_LABEL}"
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 [BatchRT.conf](#) file):

MT\_SMTP\_SERVER : SmtServer (default value : “localhost”)

MT\_SMTP\_PORT: SmtPort (default value : “25”)

### Options

**-t ‘<To-Address1>[,<To-Address2>...]’**

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.

**-a <attach-file1>[,<attach-file2>...]**

Optional.

ddname of the attached document(s).

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

**-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.

**-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 “,”.

**-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`.

ddname 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 `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 `ARTJESINITIATOR` in that server group is started, the job will be in the waiting queue until the `ARTJESINITIATOR` in that server group is started.

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

## Options

### **SvrGrpName**

Mandatory.

The JES server group name. The `ARTJESINITIATOR` deployed in this server group executes the current job.

### Example

```
m_JobBegin -j SAMPLEJCL -s START -v 2.0 -c R
m_JobSetExecLocation "ATLANTA"

while true ;
do
    m_PhaseBegin
...
m_JobEnd
```

---

The job can be submitted on any JES node, but only be executed by the `ARTJESINITIATOR` which belongs to JES's Tuxedo server group `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 1 TuxJES Servers, Commands and Utilities**

Name	Description
<a href="#">genjesprofile</a>	Generates the security profile for TuxJES system
<a href="#">genjesacl</a>	Generates the encrypted job access authorization configuration file for TuxJES system
<a href="#">gensysprofile</a>	Generates file to encrypt and store TuxJES Database connection information
<a href="#">artjesadmin</a>	TuxJES command interface.
<a href="#">artjescleanlock</a>	Cleans stale lock records in lock files in MP mode.
<a href="#">ARTJESADM</a>	TuxJES administration server.

**Table 1 TuxJES Servers, Commands and Utilities**

Name	Description
<a href="#">ARTJESCONV</a>	TuxJES conversion server.
<a href="#">ARTJESINITIATOR</a>	TuxJES job control API.
<a href="#">ARTJESPURGE</a>	Purges job queue.
<a href="#">TuxJES Queue System</a>	TuxJES Queue system.
<a href="#">TuxJES User Substitution</a>	Defines the specific OS user to execute jobs
<a href="#">TuxJES Trace Message Control</a>	Controls TuxJES Trace Messages

## genjesprofile

### Name

`genjesprofile` – Generates the security profile for TuxJES system

### Synopsis

```
genjesprofile [-f userprofile]
```

### Description

`genjesprofile` generates the TuxJES system security profile. When `genjesprofile` is launched, you are prompted to enter the Oracle Tuxedo application password, user name, user password, the database connection string for `MT_DB_LOGIN`, the database connection string for `MT_GDG_DB_ACCESS`, the database connection string for `MT_DB_LOGIN2`, and the database connection string for `MT_CATALOG_DB_LOGIN`, and the ftp password for `MT_FTP_PASS`.

The output is a security profile file that contains the Oracle Tuxedo application password, user name, user password, the database connection string for `MT_DB_LOGIN`, the database connection string for `MT_GDG_DB_ACCESS`, the database connection string for `MT_DB_LOGIN2`, and the database connection string for `MT_CATALOG_DB_LOGIN` with the file permission `-rw-----`, and the ftp password for `MT_FTP_PASS`.

**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](#)

# genjesacl

## Name

genjesacl - Generates the encrypted job access authorization configuration file for TuxJES system.

## Synopsis

```
genjesacl [-f <output_file>]
```

## Description

genjesacl reads plain rules from STDIN line by line until EOF (pressing Ctrl+D at the beginning of the line in terminal can produce EOF), and then generates an encrypted configuration file. It must be used when JES\_ACL\_FILE\_TYPE=ENCRYPTED is specified.

The generated encrypted configuration file can be used by the ARTJESADM tool to authorize TuxJES job access.

## Parameters and Options

genjesacl supports the following parameters and options:

**[-f <output\_file>]**

The location of the encrypted rule file that genjesacl generates. If this option is not specified, the default value is ~/.jesAclEncrypted.\

## See Also

[ARTJESADM](#)

## gensysprofile

### Name

`gensysprofile` – Generates file to encrypt and store TuxJES Database connection information.

### Synopsis

```
gensysprofile -d <JESROOT>
```

### Description

`gensysprofile` generates file to encrypt and store TuxJES Database connection information.

### Parameters and Options

`gensysprofile` supports the following parameters and options:

**-d <JESROOT>**

Specifies JESROOT directory, where this file is to be generated.

## artjesadmin

### Name

`artjesadmin` – TuxJES command interface.

### Synopsis

- `artjesadmin -v`
- `artjesadmin [-f [security_profile]]`
- `artjesadmin [-f [security_profile]] -p -j jobid`
- `artjesadmin [-f [security_profile]] -x showjobexec|resumejobexec|stopjobexec`  
`[[lmid=machine|grpId=groupid|grpId=groupid,srvid=serverid];...]`
- `artjesadmin [-f [security_profile]] [-t timeout(s)] [-o ejr option]`  
`[-y] -i|-I scriptfile`
- `artjesadmin [-f [security_profile]] -x settracelevel`  
`[trclvl=trace_level[, [lmid=machine|grpId=groupid|grpId=groupid,srvid=serverid];...]]`
- `artjesadmin -f [security_profile] -x setjesacl`  
`aclfile=rulepath[, aclfiletype=PLAIN|ENCRYPTED[, aclmode=MAC|DAC]]`

## Description

`artjesadmin` is the TuxJES command interface. 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.

**-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.

**-p and -j**

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

[Table 2](#) lists the exit codes for `artjesadmin` if `-p` is specified.

**Table 2 Exit Code**

Exit Code	Descriptions	Notes
0	Job is finished normally. Job status = DONE	A job is finished successfully.
1	Command execution fails.	The failure is caused by an internal error, a network error, or a syntax error.

**Table 2 Exit Code**

Exit Code	Descriptions	Notes
3	Job status = FAIL	JOB execution fails.
4	Job status = CANCEL	A job is canceled.
5	Job status = CONVING	A job is waiting for conversion.
6	Job status = EXECUTING	A job is running.
7	Job status = HOLD_CONVING	A job is in hold state without conversion.
8	Job status = HOLD_WAITING	A job is in hold state after conversion.
9	Job status = WAITING	A job is waiting for execution.
10	Job status = DISCARD	This status will occur if <code>topenqueue ()</code> fails.
11	Job status = INDOUBT	When a job is running, if JES server ARTJESINITIATOR is shutdown and then restarted, the job status will be INDOUBT.
22	Job does not exist.	N/A

Information shown in [Table 3](#) is printed to `stdout` in the following format.

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

**Table 3 Standard Output**

Output Content	Descriptions	Sample
<JOBID>	Job ID	00005097
<JOBNAME>	Job name	JOBA
<JOBSTATUS>	Job current status	DONE
<JOB RETURN CODE>	Job return code from EJR (only available if a job has finished)	C000

**Listing 1 Sample: Job has been Finished Normally**

---

```
00000002,JOBA,DONE,C0000
```

---

**Listing 2 Sample: Job is Finished but Fails**

---

```
00000002,JOBA,FAILED,U0568
```

---

**Listing 3 Sample: Job is Running**

---

```
00000002,JOBA,EXECUTING
```

---

```
-x showjobexec | resumejobexec | stopjobexec  
[[lmid=machine | grpId=groupid | grpId=groupid, srvid=serverid];...]
```

Option `-x` is specified to control ARTJESINITIATOR servers. It must be `showjobexec`, `resumejobexec`, or `stopjobexec`. For more information, see [Sub Commands](#).

```
-t <timeout>
```

Option `-t <timeout>` is specified to control the timeout threshold when submitting a job in synchronous mode, asynchronous mode, or console mode. With `-t` option, an integer can be specified to control the timeout. If no timer is specified, clients will wait forever.

- In synchronous mode, the timer is used to wait for the job (from submission to execution) to finish.
- In asynchronous mode or console mode, the timer is used to wait for the job (submission only) to finish.

```
-o ejr option
```

Option `-o ejr option` specifies the options passed to the EJR script file.

```
-y
```

Option `-y` is added to submit a job in the synchronous way. This option enables synchronous mode to wait for job end.

**Note:** To submit a job in synchronous way, in TuxJES Configuration File, you must set `EVENTPOST=A`.

[Table 4](#) lists the exit codes for `artjesadmin` if `-y` is specified.

**Table 4 Exit Code**

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

Information shown on [Table 5](#) will be printed to stdout in the following format.

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

**Table 5 Standard Output**

Output Content	Descriptions	Sample
<JOBID>	Job ID	00005097
<JOBNAME>	Job name	JOBA
<JOBSTATUS>	Job final status (only available if job is finished before the timeout occurs)	DONE
<JOB RETURN CODE>	Job return code from EJRC (only available if a job is finished before the timeout occurs)	C000

**Listing 4 Sample: Job is Executed Successfully**

```
00000002, JOBA, DONE, C0000
```



**Listing 5 Sample: Job Fails**

---

```
00000002,JOBA,FAILED,U0568
```

---

**Listing 6 Sample: Timeout Occurs**

---

```
00000002,JOBA,Already Timeout!
```

---

**-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.

**-x settracelevel**

```
[trclvl=trace_level[, [lmid=machine|grpId=groupid|grpId=groupid, srvid=serve
rid];...]]
```

Option **-x settracelevel** is specified to set the TuxJES trace message level. For more information, see [Sub Commands](#).

**-x setjesacl**

```
aclfile=rulepath[, aclfiletype=PLAIN|ENCRYPTED[, aclmode=MAC|DAC]]
```

Option **-x setjesacl** is specified to change job access authorization. For more information, see [Sub Commands](#).

## Sub Commands

artjesadmin supports the following sub commands:

```
submitjob (smj) -i|-I scriptfilename [-t timeout] [-o ejr option]
```

Submits a job to TuxJES system. The `scriptfilename` 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:

- **-i scriptfilename:** The script file.

## Tuxedo Job Enqueueing Service (TuxJES)

- `-I scriptfilename`: The option specified to submit JCL jobs.
- `-t timeout`: Specifies to control the timeout threshold when submitting a job.
- `-o ejr option`: Specifies the options passed to the EJR script file.

**showjobexec**(she) [-n machine] | [-g groupid [-i serverid]]

Display the number of executing jobs of ARTJESINITIATOR servers. If no option is specified, display the number of executing jobs of all ARTJESINITIATOR servers.

**-n machine:**

The Tuxedo logic machine name that the ARTJESINITIATOR server is running on.

**-g groupid:**

The Tuxedo group id of the ARTJESINITIATOR server.

**-i serverid:**

The Tuxedo server id of the ARTJESINITIATOR server.

**stopjobexec**(ste) [-n machine] | [-g groupid [-i serverid]]

Stops ARTJESINITIATOR servers from picking up a new job to execute; the server continue finishing current jobs. If no option is specified, all ARTJESINITIATOR servers stop picking up new jobs.

**-n machine:**

The Tuxedo logic machine name that the ARTJESINITIATOR server is running on.

**-g groupid:**

The Tuxedo group id of the ARTJESINITIATOR server.

**-i serverid:**

The Tuxedo server id of the ARTJESINITIATOR server.

**resumejobexec**(rse) [-n machine] | [-g groupid [-i serverid]]

Resume ARTJESINITIATOR servers pick up of new jobs to execute. If no option is specified, all ARTJESINITIATOR servers resume picking up new jobs.

**-n machine:**

The Tuxedo logic machine name that the ARTJESINITIATOR server is running on.

**-g groupid:**

The Tuxedo group id of the ARTJESINITIATOR server.

**-i serverid:**

The Tuxedo server id of the ARTJESINITIATOR server.

**settracelevel**(stl) -t tracelevel [-n machine] | [-g groupid [-i serverid]]

Set the TuxJES trace message level of TuxJES client and server.

If none of the `machine`, `groupid`, and `serverid` parameter is specified, the `artjesadmin` will change the TuxJES trace message level of the current client and all servers that your `UBBCONFIG SERVERS` section specifies. Once you specify one or more parameters, the `artjesadmin` will only change the TuxJES trace message level of the servers that you specifies; the current client will not be changed.

**-t tracelevel:**

Specify the TuxJES trace message level parameter. `-t tracelevel` can be set as 0, 1, 2, or 3. 0 indicates ERROR level; 1 indicates WARN level; 2 indicates INFO level; 3 indicates DEBUG level.

**-n machine:**

Specify the Tuxedo logic machine which the TuxJES server is running on.

**-g groupid:**

Specify the Tuxedo group which the TuxJES server belongs to.

**-i serverid:**

Specify the Tuxedo server ID of the TuxJES server.

**setjesacl (sja) -f rulepath [-t PLAIN|ENCRYPTED] [-m MAC|DAC]**

Dynamically change job access authorization. Only the Oracle Tuxedo user who is mapped to (or identical with) Unix/Linux root (referred to as super user) can use this sub command.

**-f rulepath:**

Specify the path of job access authorization configuration file. Its usage is just likes `JES_ACL_FILE` in `JESCONFIG`.

**-t PLAIN|ENCRYPTED:**

Specify the rule file is encrypted or not. If not specified, use the default value `PLAIN`. Its usage is just likes `JES_ACL_FILE_TYPE` in `JESCONFIG`.

**-m MAC|DAC:**

Specify the action when no matching rule is found for the user. If not specified, use its default value `MAC`. Its usage is just likes `JES_ACL_MODE` in `JESCONFIG`.

## Error(s)

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 6](#).

**Table 6 Error Codes**

Code	Description
0	No runtime error
251	artjesadmin it self command error returned by ARTJESADM server
252	JES2SUBMIT service error
Others	EJR none zero exit code

## Printing

**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

```
> ptj -a
JOBNAME  JobID   Owner    Prty C    Type     Status
-----  -
cjob     00000015 *        5 A      KSH      DONE
cjob     00000016 *        5 A      KSH      DONE
cjob     00000018 *        5 A      KSH      CONVING

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 `CONVING` or `WAITING` status. The options are as follows:

- n jobname: Hold jobs with given job name
- j jobid: Hold a particular job
- c job\_class: Hold a particular class jobs
- 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 | -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 | -c job class | -s job status | -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
- c job class: Purge jobs with given job class
- s job status: Purge jobs with given job status
- 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”.

## ARTJESADM

### Name

ARTJESADM – TuxJES Administration server.

### 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.

### JES\_ACL\_FILE

The full path of job access authorization configuration file. This file can be plain or encrypted, see [JES\\_ACL\\_FILE\\_TYPE](#) for more information.

Because the JESCONFIG file and JES\_ACL\_FILE file are based on TuxJES security mechanism, it's important to well protect these two files at the very beginning. It is strongly recommended that read/write permissions for these two files should be granted to only the user who is responsible for booting the whole TuxJES domain (normally it is the root account).

### JES\_ACL\_FILE\_TYPE

The format of JES\_ACL\_FILE file. It can be set as PLAIN or ENCRYPTED (case insensitive). PLAIN means JES\_ACL\_FILE file is plain while ENCRYPTED means this file is encrypted. The encrypted file can be generated by `genjesacl` tool. The default value is PLAIN.

### JES\_ACL\_MODE

The action when no matching rule is found for the tuple of user, operation, and job in JES\_ACL\_FILE. It can be set as MAC or DAC. MAC (Mandatory Access Control) means all operations are denied if no matching rule is found while DAC (Discretionary Access Control) means all operations are allowed if no matching

rule is found. `MAC` is the default value. If `JES_ACL_MODE` is configured to invalid values, server `ARTJESADM` cannot boot up.

#### 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.

You can specify multiple path names, delimit them with a colon (:). For example, `JOBREPOSITORY=<path1>:<path2>:<path3>`

To find job to submit, Batch Runtime searches from these paths in the order that you specify (in `JOBREPOSITORY`). When finding a job name match, Batch Runtime stops searching, and submits this matched job.

#### 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.

#### USE\_DB

Enable to use Database to store job management data. Values could be:

`ORACLE`: Use Oracle Database to store job management data.

`DB2`: Use DB2 as storage to store job information. This is currently not supported.

`NOT SET`: Use `/Q` to store job information.

## Tuxedo Job Enqueueing Service (TuxJES)

### Example(s)

UBBCONFIG example:

```
ARTJESADM
    SRVID=1 SRVGRP=SYSGRP CLOPT="-A -- -i /nfs/users/jes/jesconfig"
```

### 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 7](#).

**Table 7** TuxJES Queues

Queues	Description
JCL Conversion Queue	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 "JCL conversion queue" to the "conversion queue". The jobs in the queue are processed in FIFO order. Queue name: CONV_JCL
Conversion Queue	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 "conversion queue" to the "execution queue". The jobs in the queue are processed in FIFO order. Queue name: CONV

**Table 7 TuxJES Queues**

Queues	Description
ExecGroup Queue	<p>This queue is necessary if NJESUPPORT is enabled in <code>jesconfig</code> file. This queue maintains the mapping of each job and its execution group when the job is in the Execution Queue.</p> <p>Queue name: EXECGRP</p>
Hold Queue	<p>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 <code>typrun</code> parameter.</p> <p>Queue name: HOLD</p>
Execution Queue	<p>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.</p> <p>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.</p> <p>Queue names: [A-Z], [0-9].</p>
Executing Queue	<p>This queue stores running/executing jobs. There is only one "executing queue" in the system. When a job is picked up from an "execution queue" and successfully goes to running state, the job is moved to the "executing queue". The jobs in this queue are processed in FIFO order.</p> <p>Queue name: EXEC</p>
Output Queue	<p>When a job is completed or an error occurs, it is sent to the "output queue". There is only one "output queue". The jobs in the queue are processed in FIFO order.</p> <p>Queue name: OUTPUT</p>
Purge Queue	<p>When a job is to be purged, it is moved to the purge queue. There is only one "purge queue" in the system.</p> <p>Queue name: PURGE</p>
Internal Queues	<p>ART JES also has some internal queues on the JES2QSPACE for internal usage.</p>

## The TuxJES Queue Creation Script

The TuxJES system provides a sample shell script (`jesqinit`) to create the queue space (JES2QSPACE) and the queues listed in [Table 7](#). 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

## 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

Recommended queue space size: 4K x (number of jobs per day) x (days jobs stay in JES queues before being auto purged)

**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 8](#) lists the terms you need to know for a good understanding of the user substitution feature described in this section.

**Table 8 User Substitution Terms**

Term	Description
Job submitter	The OS user who submits a JES job with <code>artjesadmin</code>
Job owner	The Oracle Tuxedo user who is displayed as the owner of a JES job
Job user	The OS user who submits, cancels or prints a JES job with <code>artjesadmin</code>
Job executor	The OS user who executes the EJL to execute a JES job



## 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 9](#) shows the relation between job owner designation and different configuration scenarios.

**Table 9 Job Owner Designation in Different Configuration Scenarios**

Configuration Scenario		Job Owner
When <code>artjesadmin</code> is executed with...	When <code>PRIVILEGE_MODE</code> is set to...	
a profile containing a user name	NONE	Oracle Tuxedo user
a profile containing a user name	<code>USER_IDENTICAL</code> or <code>USER_MAPPING</code>	Oracle Tuxedo user <b>Note:</b> Oracle Tuxedo <code>SECURITY</code> parameter must be set to <code>USER_AUTH</code> , <code>ACL</code> or <code>MANDATORY_ACL</code> , otherwise JES system fails to start up and the following error message is printed into <code>ULOG</code> : "ERROR: The current Tuxedo security level disallow the given privilege mode".
no profile or a profile containing no user name	NONE	* A job with an owner "*" can be controlled by anyone.
no profile or a profile containing no user name	<code>USER_IDENTICAL</code> or <code>USER_MAPPING</code>	<code>artjesadmin</code> fails to login and the following error message is displayed: "ERROR: failed to join application."

## Permissions of Job Operations

[Table 10](#) and [Table 11](#) show the permissions of users with different identities when operating jobs in TuxJES system enabling the user substitution.

**Table 10 Permissions of Job Operations When PRIVILEGE\_MODE is set to "USER\_IDENTICAL"**

User	Permission
Oracle Tuxedo user "root"	print, hold, release, cancel, purge, and control jobs
Owner of the job	print, hold, release, cancel, purge, and control jobs
Oracle Tuxedo users who have the read and execution permissions to the corresponding job directory	print jobs

**Table 11 Permissions of Job Operations When PRIVILEGE\_MODE is set to "USER\_MAPPING"**

User	Permission
Oracle Tuxedo user who is mapped to OS user "root"	print, hold, release, cancel, purge, and control jobs
Oracle Tuxedo user who is mapped to job executer	print, hold, release, cancel, purge, and control jobs
Oracle Tuxedo user who is mapped to a OS user with read and execution permissions to the corresponding job directory	print jobs

## 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

---

```
#      (c) 2011 Oracle. All Rights Reserved.
#
*RESOURCES
IPCKEY      <IPCKEY> # for example 132770
DOMAINID    jessample
MASTER      SITE1
MODEL       SHM
MAXACCESSERS 200
```

## Tuxedo Job Enqueueing Service (TuxJES)

```
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"

```

## Tuxedo Job Enqueueing Service (TuxJES)

```
RQPERM=0666 RPPERM=0666
```

```
ARTJESPURGE      SRVGRP =ARTGRP  SRVID = 100
```

```
CLOPT = "-A --"
```

```
AUTHSVR SRVGRP=ARTGRP SRVID=104 CLOPT="-A"
```

```
RQPERM=0666 RPPERM=0666
```

```
*SERVICES
```

---

### 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 Trace Message Control

TuxJES trace messages are stored in TuxJES trace file. By default, TuxJES trace file is stored in `${APPDIR}/Logs` directory; before running Batch Runtime, you can change the directory by setting the environment variable `JES_TRACE_PATH`.

There are four TuxJES trace messages levels: `ERROR`, `WARN`, `INFO`, and `DEBUG`. `WARN` is used by default. You can use environment variable `JESTRACE` to set the TuxJES trace message level, or use command `artjesadmin` to dynamically change it, determining which level of messages will be displayed.

For more information, see [Tracing TuxJES](#) in *Oracle Tuxedo Application Runtime for Batch User Guide* and [artjesadmin](#) in *Oracle Tuxedo Application Runtime for Batch Reference Guide*.

## 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 12](#) illustrates the fields used in job operation.

**Table 12 FML32 Fields in Use for TuxJES Job Operation**

Field Name	Type	Descriptions
<code>JES2_JOB_ID</code>	<code>STRING</code>	The job ID generated by TuxJES job management if a job is successfully submitted to the system. The range is [1-99999999].
<code>JES2_JOB_NAME</code>	<code>STRING</code>	The job name given to this job. It is supplied by users in the shell script.

**Table 12 FML32 Fields in Use for TuxJES Job Operation**

<b>Field Name</b>	<b>Type</b>	<b>Descriptions</b>
JES2_JOB_CLASS	STRING	The job class.
JES2_JOB_PRTY	SHORT	The job priority ranged in [0-15].
JES2_JOB_TYPRUN	STRING	The job type. It is supplied by users for job submission in the shell script.
JES2_JOB_SCRIPT	STRING	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.
JES2_JOB_EJROPTION	STRING	The options passed to EJRS itself.
JES2_JOB_OWNER	STRING	The job owner. It is supplied when a job is submitted and returned by server to query job information.
JES2_JOB_SUBMITTIME	LONG	The submitted timestamp of a job. It is the seconds since Epoch time.
JES2_JOB_STATUSES	STRING	The job status returned by server.
JES2_JOB_ERROR	STRING	If the operation encounters some errors, the field carries the information.
JES2_JOB_MSG	STRING	It carries error information for particular jobs. Unlike JES2_JOB_ERROR, JES2_JOB_MSG carries general failure information.
JES2_JOB_CURRENT	STRING	The current step of a job.
JES2_JOB_ENDTIME	LONG	The timestamp of a terminated job. It is seconds since Epoch time.
JES2_JOB_EXECUTEIME	LONG	The timestamp of an executing job. It is seconds since Epoch time.
JES2_JOB_EXECUTEAC	STRING	The host for the running job.
JES2_JOB_USERSEC	LONG	The user-level CPU time in seconds.



**Table 12 FML32 Fields in Use for TuxJES Job Operation**

Field Name	Type	Descriptions
JES2_JOB_USRUS EC	LONG	The user-level CPU time for the microseconds portion in one second.
JES2_JOB_SYSS EC	LONG	The system-level CPU time in seconds.
JES2_JOB_SYSUS EC	LONG	The system-level CPU time for the microseconds portion in one second.

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. 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 tperno 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\_SUBMITTIME
- JES2\_JOB\_CLASS
- JES2\_JOB\_TYPRUN
- JES2\_JOB\_PRTY
- JES2\_JOB\_OWNER
- JES2\_JOB\_EXECMAC
- JES2\_JOB\_USRSEC
- JES2\_JOB\_USRUSEC
- JES2\_JOB\_SYSSEC
- JSE2\_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.

**Table 13 Available TuxJES Events for Subscription**

Event Name	Descriptions
ARTJES_JOBSUB MIT	A new job is submitted.
ARTJES_JOBCVT	A job conversion is completed.
ARTJES_JOBCVT _FAIL	A job conversion fails.
ARTJES_JOBEXE C	A job is finished successfully.
ARTJES_JOBEXE C_FAIL	A job's execution fails.
ARTJES_JOBPUR GE	A job is purged.
ARTJES_JOBCAN CEL	A job is canceled.

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`.

### Listing 9 Sample of submitting a job

---

```
#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", tpsterror(tperrno));
        exit(1);
    }
}
```



```

input = tmalloc("FML32",NULL,1024);
output = tmalloc("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);
}

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);
    }
}

tpfree(input);
tpfree(output);
(void)tpterm();
return(0);
}

```

## Tuxedo Job Enqueueing Service (TuxJES)

---