

# UniKix Batch Processing Environment Software Solution Brief

UniKix™ Batch Processing Environment (BPE) software provides a comprehensive, native framework for administering, executing, and managing batch workloads on open systems – bringing the best aspects of mainframe job management to cost-effective, open systems. With comprehensive Job Control Language (JCL) translators, pre-built integration with common mainframe and 3<sup>rd</sup> party batch utilities, and native support for standard mainframe data file types, UniKix BPE complex integration. A mature and robust solution, UniKix BPE software is backed by over 16 years of experience, detailed documentation guides, and worldwide support. Compared to alternative migration approaches which re-implement business procedures and environmental variables, UniKix BPE software from Clarity Solutions, Inc. (Clarity) delivers affordable, code-efficient, and highly-manageable production environments.

UniKix BPE software customers benefit before, during, and after a migration by opting to deploy this robust environment. Key advantages of UniKix BPE technology over alternatives commonly include the following:

- **Flexible and evolutionary path forward**  
UniKix BPE software can be leveraged to quickly and efficiently migrate COBOL, PL/1 and related application code; organizations can choose to continue programming in JCL or program in standardized shell scripts moving forward.
- **Streamlined, maintainable source code through comprehensive JCL translators**  
UniKix BPE software includes time-tested and proven IBM® z/OS® environment and IBM® z/VSE® environment translators that deliver streamlined, manageable solution code and reduce migration time.
- **Built-in integration with common IBM mainframe utilities, 3<sup>rd</sup> Party utilities, and JCL syntax**  
Pre-built integration with application utilities and JCL syntax enables UniKix BPE technology to automate a large percentage of the integration efforts commonly associated with migrating 'utility-rich' (i.e. sorting, printing, reporting,...) batch environments.
- **Native support for standard mainframe data file types**  
With native support for sequential files, GDGs for COBOL, Concatenated Data Sets, and other standard mainframe data types, migrating with UniKix BPE software means less data conversion work and fewer changes to customer code during migrations.
- **Robust management and reporting functionality; support for job prioritization and classes**  
UniKix BPE software provides a familiar way to get what is needed in terms of process tracking, monitoring, and reporting. This functionality does not have to be reengineered during the migration.
- **Well-documented environment with easy to use GUI and command line interface**  
A complete documentation set and choice of administration vehicles confidently equips administrators to manage a UniKix BPE software environment after a migration is complete.
- **Comprehensive, worldwide support options**  
Organizations with stringent Service Level Agreements (SLAs) agreements can select the support level agreement appropriate for their particular business requirements.

The rest of this solution brief discusses UniKix BPE software advantages in more detail.

## 1. Flexible, evolutionary path forward

UniKix BPE software offers customers the option of continuing to program in JCL syntax or to translate JCL to a UniKix BPE technology scripting language. With UniKix BPE technology, JCL on a distributed environment is translated to scripting language and then executed later in the target platform. A customer is not bound to the JCL and, if desired, can discontinue using JCL syntax after a migration is complete. With many customers, the capability to continue utilizing time-tested JCL is welcomed as it allows for the preservation of valuable and irreplaceable human skills, however, shell scripts can run within a UniKix BPE region.

## 2. Streamlined, maintainable source code through comprehensive JCL translators

JCL was originally developed on mainframes to provide a comprehensive framework in which to manage and prioritize batch job tasks. When an alternative approach is used to migrate JCL code that re-implements procedures and variables, 5 to 10 times the number of extra lines of code (LOC) may be generated to address functionality and environmental variables previously handled in JCL. With UniKix BPE software, there is no need to re-implement complex batch processes and procedures when mainframe workloads are migrated to open systems.

Over the past decade, UniKix BPE software has honed a native environment to address functionality delivered within JCL Syntax. IBM z/OS environment and IBM z/VSE environment JCL code is translated with UniKix BPE software tools, instead of being re-implemented. Compared to approaches which re-implement JCL functionality, utilities, and environmental variables in scripts during a migration -- increasing the volume of directives a customer needs to maintain -- Clarity's approach delivers streamlined and manageable code on the new target platform.

By replacing common JCL syntax on a one-for-one code line basis, application environments can be efficiently migrated. Post-migration, when an environmental variable needs to be changed, administrators can make the required adjustments once from the UniKix BPE software administration tool console instead of having to go through all lines of application code and make required changes everywhere a particular variable is referenced.

### JCL code example

A migration approach that re-implements JCL utilities and variables can take the simplest job and turn it into a significant coding and maintenance challenge. Here a simple job "JOB00289" is first run on a mainframe system. As shown in Table 1 below, "Comparison of IBM z/OS Environment, UniKix BPE Software, and Re-Implemented Translation," this activity takes 35 LOC in the original IBM z/OS environment JCL, 33 LOC in the rehosted UniKix BPE software environment, and 244 LOC using an alternative approach. Nearly 600% more code needs to be generated in order to complete this solution with the alternative, re-implementation approach.

Table 1. Comparison of IBM z/OS Environment, UniKix BPE Software, and Re-Implemented Translation\*

	Original IBM z/OS environment JCL	JCL Translated with UniKix BPE software	Re-implemented approach
Total Lines Of Code (LOC)*	35	33	244
% LOC versus original JCL	100% (same)	95%**	597%

\*In these examples, actual source code lines were counted – not comments, spaces etc.

\*\*UniKix BPE software results in few LOC than the original JCL because it consolidates lines when executing a procedure.

JOB00289 is a four step JCL program executing the following activity:

- The first step defines where programs are to be retrieved, then executes an IDCAMS step.
- In the second step, a procedure passes some parameters which are overridden in the JCL.
- The third step uses IEBGENER to copy output from the previous step to the internal reader.
- The fourth and final step executes a COBOL program passing parameters, with a GDG, VSAM, and sequential file used as input.

As illustrated in Table 2, "Original MVS JCL," and in the related tables found in Appendix 1, "JCL Migrated with UniKix BPE Technology" & Appendix 2, "JCL Converted with Re-Implemented Scripts," a large portion of the code in the re-implemented example has been generated to address required environmental variables -- functionality that is present out of the box with UniKix BPE software.

Table 2. Original MVS JCL

#### Job 00289

```
//JOB00289 JOB JF21,MSGCLASS=J,CLASS=H,
// REGION=4M
//JOBLIB DD DISP=SHR,DSN=TFHT.GEN1.TEMP.LOADLIB
// DD DISP=SHR,DSN=TFTPROD.BATCH.LOADLIB
// DD DISP=SHR,DSN=TFHT.LOADLIB
//*
/*****
//STEP1 EXEC PGM=IDCAMS
/*****
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
DELETE TFHT.FG1.INTERNAL.READ0289
SET MAXCC=0
/*
/*****
//STEP2 EXEC TFTPDISP3,
/*****
// REGION=20M,
// AR1=TFTVPRD,
// AR2=TFTVPRD,
// PARM1='N1332034500999TFTVPRDTFTVPRD',
// SYSOUT=A
/*
//STEP010.INTREAD DD DSN=TFHT.GEN.INTERNAL.READ0289,
// DISP=(,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(TRK,(01,10),RLSE),
// RECFM=FB,LRECL=80
/*
/*****
//STEP3 EXEC PGM=IEBGENER
/*****
//SYSIN DD DUMMY
//SYSUT1 DD DSN=TFHT.GEN.INTERNAL.READ0289,DISP=SHR
//SYSUT2 DD SYSOUT=(A,INTRDR)
/*****
/* ONLY STEP FOR PRIMARY AND BACKUP CONTROL FILES
/*****
//STEP4 EXEC PGM=FCHPGM1,PARM='CA7STAT',COND=EVEN,REGION=2048K
/*****
//FCHSITE DD DSN=FCHVP.SITE.TABLE,DISP=SHR,AMP='AMORG'
//FCHCCFP DD DSN=FCHVP.CONTROL.PRIMARY,DISP=SHR,AMP='AMORG'
//FCHCCFB DD DSN=FCFCH.CONTROL.BACKUP,DISP=SHR,AMP='AMORG'
//STATPRT DD SYSOUT=9,DEST=RMT63,FCB=DTRK
//PRINT DD SYSOUT=9,DEST=RMT62,FCB=DTRK
//PRINT2 DD DUMMY
/*
```

For examples of corresponding UniKix BPE technology translated JCL and the code generated in a re-implemented approach, see Appendix 1 and Appendix 2.

### 3. Built-in integration with common IBM mainframe and 3<sup>rd</sup> Party utilities

A large percentage of mainframe batch workload processing time involves interacting with common IBM mainframe utilities such as IDCAMS, IEBGENER, and IEBCOPY, in addition to 3<sup>rd</sup> party solutions such as sort utilities, job schedulers, and print spoolers. With built-in integration for both tightly and loosely coupled application utilities, UniKix BPE software automates a large portion of the integration efforts commonly associated with migrating batch environments to open systems.

Database connectivity and sort utilities are often tightly coupled to mainframe application code. Thus, support for such activity needs to be built in at runtime to run efficiently. UniKix BPE software, being a complete batch environment, supports databases and sort utilities out of the box using optimized integration options. Loosely coupled activities such as print spooling and job scheduling also consume large portions of mainframe compute cycles. UniKix BPE software provides seamless support for related job scheduling and print spooling solutions. UniKix BPE software provides post-exec scripts for all major print spoolers including Levi, Ray & Shoup VPSX, Macro4 Columbus OM, Isis Papyrus, Open Text Vista Plus, and LBM Systems UniQue. UniKix BPE software has integrated scripts for all major distributed job schedulers, including BMC Control-M, CA-Autosys, IBM® Tivoli® Workload Scheduler, Tidal Enterprise Job Scheduler, and Dollar Universe. UniKix BPE is extensible to support other print distribution facilities or job schedulers through standard user exit procedures.

The script extract provided in Appendix 3, “Example UniKix BPE software and Levi, Ray and Shoup VPSX Script Extract,” is just a portion of an example post-exec script supplied with UniKix BPE software. This script provides integration between the UniKix BPE environment and the VPSX print output management solution from Levi, Ray and Shoup, Inc. (LRS). This example details the complex, pre-built scripts that have already been developed for UniKix BPE software to address loosely-coupled utility requirements; functionality that would need to be recreated from scratch with a re-implemented approach.

Not only does UniKix BPE technology's native support for mainframe utilities and third party software enable efficient migrations, the high level of integration with IBM mainframe and 3<sup>rd</sup> Party utilities built into UniKix BPE technology also enables Independent Software Vendors (ISVs) looking to deploy their business applications to their customers on mainframes and open systems to maintain one code base for applications on both platforms going forward.

### 4. Native support for standard mainframe data file types

UniKix BPE software supports standard mainframe data file types such as sequential files, GDGs (Generation Data Groups), and Concatenated Data Sets. Thus, manual conversion work in this space is minimized.

If one implements a solution that does not support data file types such as GDGs and concatenated data sets and utilizes applications designed to use these features, a migration cannot be completed without either re-implementing current source code or reinventing such functionality on the new target platform. Without the logic found in UniKix BPE software to address these data file types, data file iterations may be collected into one large file. This could potentially consume more disk space and compute cycles on a target system since the data would need to be handled during every job step attempting to read these files.

### 5. Robust management and reporting functionality; support for job prioritization and classes

Compared to a re-implemented approach which lacks a complete batch environment to interface with, UniKix BPE software provides a comprehensive and familiar way to get what is needed in terms of process tracking, monitoring, and reporting.

As illustrated in Figure 1, “UniKix BPE Software Complete Jobs Records Screenshot,” UniKix BPE software includes a powerful monitoring, reporting, and administration environment that does not exist with alternatives. UniKix BPE technology also provides default accounting information about jobs, as illustrated in Figure 2, “UniKix BPE Software Job Listing Screenshot.” This information can be tailored and fed into chargeback solutions such as IT Charge Management from SAS. UniKix BPE is

extensible to provide alert, state, and status information to popular system management facilities, such as BMC Patrol, CA UniCenter or Compuware Proxima Centauri.

Figure 1. UniKix BPE Software Completed Jobs Records Screenshot

Job	Job Name	Status	User	Ended At	Size
59	FOC01MFR	Canceled	mtpadm	2007/04/03.15:11:15	532
58	FOC01MFR	Terminated	mtpadm	2007/04/03.15:08:22	2262
57	FOC01MFR	Terminated	mtpadm	2007/04/03.14:41:49	2262
56	FOC02MFR	Aborted	mtpadm	2007/04/03.14:36:50	1997
55	FOC01MFR	Terminated	mtpadm	2007/04/03.11:23:28	2262
53	FOC01MFR	Canceled	mtpadm	2007/03/21.16:04:07	532
52	FOC01MFR	Terminated	mtpadm	2007/03/21.16:01:12	2262
51	FOC01MFR	Terminated	mtpadm	2007/03/20.09:55:47	2262
50	FOC01MFR	Terminated	mtpadm	2007/03/20.09:53:29	2262
49	FOC01MFR	Terminated	mtpadm	2007/03/20.09:37:29	2262
48	FOC02MFR	Aborted	mtpadm	2007/03/20.09:32:50	1997
47	FOC01MFR	Terminated	mtpadm	2007/03/19.16:39:48	2262
46	FOC02MFR	Aborted	mtpadm	2007/03/19.16:35:55	1997
45	FOC01MFR	Terminated	mtpadm	2007/03/19.16:30:49	2262
44	FOC01MFR	Terminated	mtpadm	2007/03/15.08:50:00	2262
43	FOC02MFR	Aborted	mtpadm	2007/03/15.08:48:17	1997
42	FOC01MFR	Terminated	mtpadm	2007/03/02.10:29:38	2262
41	FOC01MFR	Terminated	mtpadm	2007/03/02.10:23:54	2262
40	FOC01MFR	Terminated	mtpadm	2007/03/02.10:09:55	2262
39	FOC02MFR	Aborted	mtpadm	2007/03/02.10:05:51	1997

This UniKix BPE software screenshot shows job information from an active job log. Information about these jobs can be queried and analyzed from a single GUI window.

Figure 2. UniKix BPE Software Job Listing Screenshot

```

Job Name      FOC01MFR
Job Number   58
Elapsed Time  19.85 (sec)
CPU Time     0.91 (sec)
User Time    0.28 (sec)
System Time  0.63 (sec)
Started at   2007/04/03.15:08:02
Ended at    2007/04/03.15:08:22
User Name   mtpadm
Job Class   a
Job Status  Terminated
Subsystem Name mFrracu

FOCUS 6.9 <UNIX> CREATED 10/21/01 6107.10
>>>>>>

NUMBER OF RECORDS IN TABLE= 175 LINES= 88
FOC01MFR(58) LFD630 STEP0001 MA2517 (I) starting
(I) /tmp/sysoutdir/FOC01MFR/JS0010_STEP0001/PRNT001_58 SYSOUT

VPSX interface starting at: Tue 03 Apr 2007 03:08:21 PM MST
SYSOUT DDNAME PRNT001
  dest      : PHX03P1
  forms     : <default>
  lines     : <default>
  copies    : 1
  priority  : <default>
  filename  : /tmp/sysoutdir/FOC01MFR/JS0010_STEP0001/PRNT001_58
Successful Print Request

Number of outputs generated for ddname PRNT001 : 1
  
```

This UniKix BPE software screenshot provides details about the information available on particular jobs. This information can be feed into accounting and reporting software as well as other statistical packages.

UniKix BPE software provides support for report generation languages and utilities such as FOCUS from Information Builders, SAS, and CA-Easytrieve Plus. This functionality does not have to be re-implemented as part of a migration project with UniKix BPE software.

Support is also provided in UniKix BPE software for common batch features such job prioritization and classes. For instance, from the administration console a system administrator may assign threads to run high priority workloads.

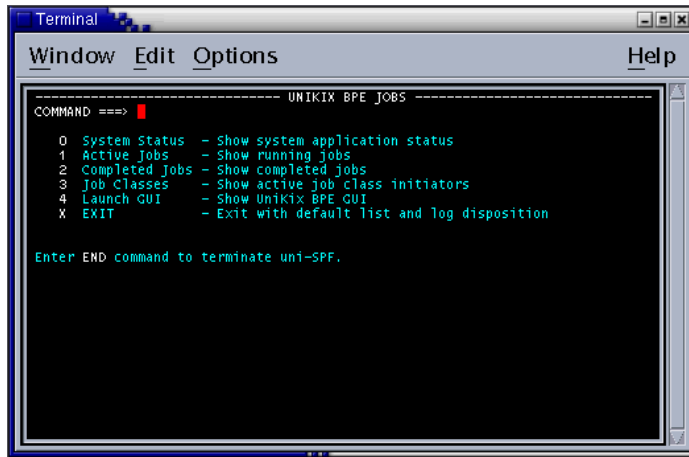
## 6. Familiar, documented environment with easy to use GUI and command line interface

UniKix BPE software has the benefit of over a decade of deployment and implementation experience. UniKix BPE software administrators are equipped with a complete Graphical User Interface (GUI) and a Command Line Interface (CLI), as well as a robust set of product documentation. As shown in Figures 3 and 4, UniKix BPE software is also works with The Workstation

Group's uni-SPF solution. The uni-SPF product provides an ISPF interface which will be familiar to users with mainframe experience.

For administrators, there is less disruption and training time required to become comfortable working with UniKix BPE technology, since it is a known framework. Administrators benefit from the documents, training, and support services Clerity personnel offer and are able to learn this environment from time-tested training courses instead of customized, one-off manuals or guidelines.

Figure 3. UniKix BPE Software and The Workstation Group's uni-SPF



```

Terminal
Window Edit Options Help
----- UNIKIX BPE JOBS -----
COMMAND ==>
0 System Status - Show system application status
1 Active Jobs - Show running jobs
2 Completed Jobs - Show completed jobs
3 Job Classes - Show active job class initiators
4 Launch GUI - Show Unikix BPE GUI
X EXIT - Exit with default list and log disposition

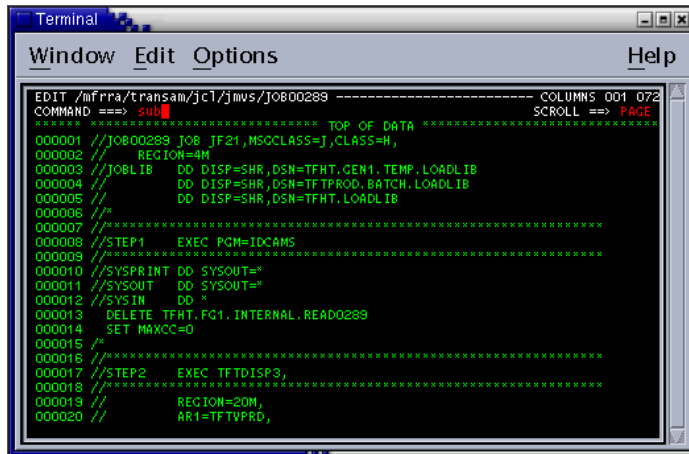
Enter END command to terminate uni-SPF.

```

This screenshot demonstrates the interaction between UniKix BPE and The Workstation Group's uni-SPF solution.

uni-SPF provides a familiar environment for mainframe personnel moving to open systems - resulting in less retraining and impact to productivity.

Figure 4. UniKix BPE Software Dynamic Job Translation



```

Terminal
Window Edit Options Help
EDIT /mfrra/transam/jc1/jmvs/JO800289 ----- COLUMNS 001 072
COMMAND ==> sub SCROLL ==> PAGE
***** TOP OF DATA *****
000001 //JOB00289 JOB JF21,MSOCLASS=J,CLASS=H,
000002 // REGION=4M
000003 //JOB LIB DD DISP=SHR,DSN=TFHT.GEM1.TEMP.LOADLIB
000004 // DD DISP=SHR,DSN=TFHTPROD.BATCH.LOADLIB
000005 // DD DISP=SHR,DSN=TFHT.LOADLIB
000006 //
000007 //*****
000008 //STEP1 EXEC PGM=IDCAMS
000009 //*****
000010 //SYSPRINT DD SYSOUT=*
000011 //SYSOUT DD SYSOUT=*
000012 //SYS IN DD *
000013 DELETE TFHT.FC1.INTERNAL.READ0289
000014 SET MMAXC=0
000015 //
000016 //*****
000017 //STEP2 EXEC TFTDISP3,
000018 //
000019 // REGION=20M,
000020 // AR1=TFTVPRD,

```

Users have the ability to submit a job to UniKix BPE software from uni-SPF. The job is dynamically translated to a UniKix BPE supported script and submitted for processing.

As listed in Table 5, "UniKix BPE Software Documentation List," versions of UniKix BPE software are supported by a full documentation suite including Installation Guides, Configuration Guides, Migration Guides, User Guides, Reference Guides, Software Message Guides and Release Notes. This eliminates guesswork on the part of administrators and programmers as to the details and management of the UniKix open systems software environment.

Table 5. UniKix BPE Software Documentation List

Document Title	Contents
UniKix BPE Software Configuration Guide	<p>Instructions for:</p> <ul style="list-style-type: none"> <li>• Configuring a node</li> <li>• Changing the date and time</li> <li>• Configuring internode communications</li> <li>• Configuring job accounting</li> <li>• Creating and customizing a subsystem</li> <li>• Configuring security and users</li> <li>• Configuring job classes and activities</li> <li>• Configuring error logging and using the troubleshooting tools</li> <li>• Customizing applications for an RDBMS</li> <li>• Communicating with UniKix TPE software</li> </ul> <p>Additional information:</p> <ul style="list-style-type: none"> <li>• Concepts</li> <li>• Environment files that control the job execution environment</li> </ul>
UniKix BPE Software Installation Guide	<p>Instructions for:</p> <ul style="list-style-type: none"> <li>• Installing UniKix BPE software</li> <li>• Upgrading from earlier releases</li> <li>• Running a sample application to verify installation</li> <li>• Setting up the license file</li> </ul>
UniKix BPE Software Message Guide	Listing of all error messages, error descriptions and suggested remedial actions
UniKix BPE Software Migration Guide	<p>Instructions for:</p> <ul style="list-style-type: none"> <li>• Mapping files</li> <li>• Using the UniKix BPE software Tool Kit to translate and validate JCL and to compile programs</li> <li>• Translating IBM® z/OS® environment and IBM® z/VSE® environment JCL</li> <li>• Using the job editor to import IBM z/OS environment and IBM z/VSE environment JCL</li> <li>• Migrating applications that use mainframe utilities</li> <li>• Migrating application programs</li> </ul> <p>Additional information:</p> <ul style="list-style-type: none"> <li>• Conceptual information about migrating jobs from mainframe environments</li> <li>• Batch shell</li> </ul>
UniKix BPE Software Reference Guide	<p>Information about:</p> <ul style="list-style-type: none"> <li>• Support for IBM z/OS environment JCL statements</li> <li>• Support for IBM z/VSE environment JCL statements</li> </ul>
UniKix BPE Software User's Guide	<p>Information and procedures for</p> <ul style="list-style-type: none"> <li>• Managing jobs and subsystems</li> <li>• Using the job editor to create and edit jobs and procedures</li> </ul> <p>Additional information:</p> <ul style="list-style-type: none"> <li>• UniKix BPE software commands</li> <li>• UniKix BPE software services and daemons</li> </ul>

## 7. Comprehensive, Worldwide Support Options

Unlike some alternatives, UniKix BPE software has flexible, worldwide support options. Software support contracts can be arranged on a one year, two year, or three year basis. Standard service support includes patches, software upgrades, and documentation. For sites with the highest Service Level Agreements (SLAs) requirements, Platinum support can also be offered which includes 2 hour response time for Priority 1 calls 24 hours a day, 7 days a week.

Additionally, access to Clerity's Support website is granted with all support contracts. This site provides an easy-to-use interface to request assistance and upload files for use by the Clerity Support Engineering Team.

### **Why Leverage UniKix BPE software?**

Given today's emphasis on "doing more with less" and the increased data center management complexities stemming from ever-growing data sources, client types, regulations, and integration requirements, the importance of effective batch workload management on open systems has never been greater. Efficient batch processing reduces time-critical latencies between "islands" of applications, minimizes the down time that can occur when a job stops or other unplanned event occurs, and automates key activities to free up administrators to deal with other tasks in the data center.

When considering a migration project, both the path to obtain the desired target environment and the destination are equally as important. With UniKix BPE software, batch application investments can be quickly be leveraged on open systems without excessive code change and complex integration efforts, resulting in a highly manageable and well-supported platform moving forward.

# Appendix

## Appendix 1

*JCL Migrated with UniKix BPE Technology*

### Job 00289

```

BEGINJOB mode='MVS' jobclass='J'

LIBDEF scope='JOB' type='PGM' dataset='TFHT.GEN1.TEMP.LOADLIB' lib='\$SEQFILES/TFHT/gen1/temp/loadlib'
LIBDEF scope='JOB' type='PGM' concat='Y' dataset='TFTPROD.BATCH.LOADLIB' lib='\$SEQFILES/tftprod/batch/loadlib'
LIBDEF scope='JOB' type='PGM' concat='Y' dataset='TFHT.LOADLIB' lib='\$SEQFILES/tfht/loadlib'

#####
LABEL name=STEP1
#####

ASSGNDD ddname='SYSRINT' type='SYSOUT' class='JOBCLASS'
ASSGNDD ddname='SYSOUT' type='SYSOUT' class='JOBCLASS'
ASSGNDD ddname='SYSIN' type='INSTREAM' << !
DELETE TFHT.FG1.INTERNAL.READ0289
SET MAXCC=0
!

EXECPGM pgmname='IDCAMS' stepname='STEP1'

#####
LABEL name=STEP2
#####

ASSGNDD ddname='STEP010.INTREAD' dataset='TFHT.GEN.INTERNAL.READ0289'
filename='\$SEQFILES/tfht/gen/internal/read0289' disp='o' normal='k' abend='d' recfmt='F' reccsize='80'

EXECPROC procname='TFTDISP3' stepname='STEP2'
parms='AR1=TFTVPRD,AR2=TFTVPRD,PARM1=N13320345009999TFTVPRDTFTVPRD,SYSOUTA=A'

#####
LABEL name=STEP3
#####

ASSGNDD ddname='SYSIN' type='DUMMY'
ASSGNDD ddname='SYSUT1' dataset='TFHT.GEN.INTERNAL.READ0289' filename='\$SEQFILES/tfht/gen/internal/read0289'
disp='i-o'
SETPRINT ddname='SYSUT2' writer='INTRDR'
ASSGNDD ddname='SYSUT2' type='SYSOUT' class='A'

EXECPGM pgmname='IEBGENER' stepname='STEP3'

# *****
# * ONLY STEP FOR PRIMARY AND BACKUP CONTROL FILES
# *****

#####
LABEL name=STEP4
#####
ONCONDDCODE NE 0 CONTINUE scope='STEP'

ASSGNDD ddname='FCHSITE' dataset='FCHVP.SITE.TABLE' filename='\$SEQFILES/fchvp/site/table' disp='i-o'
ASSGNDD ddname='FCHCCFP' dataset='FCHVP.CONTROL.PRIMARY' filename='\$SEQFILES/fchvp/control/primary' disp='i-o'
ASSGNDD ddname='FCHCCFB' dataset='FCFCH.CONTROL.BACKUP' filename='\$SEQFILES/fcfch/control/backup' disp='i-o'
SETPRINT ddname='STATPRT' dest='RMT63' fcb='DTRK'
ASSGNDD ddname='STATPRT' type='SYSOUT' class='9'
SETPRINT ddname='PRINT' dest='RMT62' fcb='DTRK'
ASSGNDD ddname='PRINT' type='SYSOUT' class='9'
ASSGNDD ddname='PRINT2' type='DUMMY'
EXECPGM pgmname='FCHPGM1' stepname='STEP4' parm='CA7STAT'
ENDJOB

```

## Appendix 2

### JCL Converted with Re-implemented Scripts

#### Job 00289

```

Start
setenv MASTER_CAT MASTERCAT
setenv COBPATH $MFERRA/transam_acu/src/batch
setenv PROCLIB /mferra/transam_acu/jcl/ishp
setenv JOB_CAT $MASTER_CAT
setenv ORIG_COBPATH $COBPATH
if ( ! -d ${SYSOUTDIR}/${JOBNAME} ) then
    mkdir -p ${SYSOUTDIR}/${JOBNAME}
endif
cat <$DEVNULL > ${JON}_WK4n
cat <$DEVNULL > ${JON}_WK4a
cat <$DEVNULL > ${JON}_RETCOD
echo "_iffirststepisproc_ 0" > ${JON}_RET
cat <$DEVNULL > ${JON}_JCOND
echo "echo 0" > ${JON}_JCOND
chmod +x ${JON}_JCOND
setenv MSGCLASS J
setenv JOBSTART "JOB $JOBNAME start at TIME=`ebmtime`"
echo "starting JOB -- JOB00289"
setenv COBPATH $SEQFILES/tfht/gen1/temp/loadlib:$SEQFILES/tftprod/batch/loadlib:$SEQFILES/tfht/loadlib:$COBPATH
setenv FILEMAP /mferra/products/MBM10.1.0p2/bam/subsys/mfrracu/File_Map
goto $RESTART
BeginJob:

#***** begin step STEP1 *****
STEP1:
#//STEP1 EXEC PGM=IDCAMS
setenv STEPNAME STEP1
setenv JOB_COND 0
echo "_${STEPNAME}_ 0" > ${JON}_RET
if ( $JOB_COND == 1 ) then
    echo "step $STEPNAME bypassed"
    echo "echo 1" > ${JON}_JCOND
else
    if ( $JOBSTATUS == 0 ) then
        setenv START "$STEPNAME start at TIME=`ebmtime`"
        echo "starting STEP -- $STEPNAME"
        setenv S_CKPPFILE 0
        cat <$DEVNULL > ${JON}_WK1
        cat <$DEVNULL > ${JON}_WK2a
        cat <$DEVNULL > ${JON}_WK2n
    cat > ./SYSIN.${JON} << !
        DELETE TFHT.FG1.INTERNAL.READ0289
        SET MAXCC=0
    !
        setenv DD_SYSIN ./SYSIN.${JON}
        echo "rm -f ./SYSIN.${JON}" >> ${JON}_WK2n
        echo "rm -f ./SYSIN.${JON}" >> ${JON}_WK2a
        echo 256 > status.${JON}
        IDCAMS
        setenv STATUS `cat status.${JON}`
        unsetenv DD_SYSIN
        if ( $STATUS == 0 ) then
            source ${JON}_WK2n
        else
            source ${JON}_WK2a
            setenv JOBSTATUS 1
        endif
        echo $STEPNAME step was executed cond code $STATUS
        echo 'step $START'
        echo 'step $STEPNAME stop\ \ at TIME=`ebmtime`'
    else
        echo "step $STEPNAME is bypassed"
    endif
endif
cat ${JON}_RET >> ${JON}_RETCOD
unsetenv PARM
#***** end step STEP1 *****

#***** begin step STEP2 *****

```

```

STEP2:
#//STEP2   EXEC TFTDISP3,REGION=20M,
#//       AR1=TFTVPRD,
#//       AR2=TFTVPRD,
#//       PARM1='N13320345009999TFTVPRDTFTVPRD',
#//       SYSOUTA=A
setenv STEPNAME STEP2
setenv JOB_COND `${JON}_JCOND`
if ( $JOB_COND == 1 ) then
    echo "step $STEPNAME bypassed"
    echo "echo 1" > ${JON}_JCOND
else
    if ( $JOBSTATUS == 0 ) then
        setenv START "$STEPNAME start at TIME=`ebmtime`"
        echo "starting STEP -- $STEPNAME"
        setenv S_CKPFIL 0
        cat <$DEVNULL > ${JON}_WK1
        cat <$DEVNULL > ${JON}_WK3a
        cat <$DEVNULL > ${JON}_WK3n
        setenv PROCNAME STEP2
        setenv LRECL_STEP010_INTREAD 80
        rmdispnew $SEQFILES/tfht/gen/internal/read0289
        touch $SEQFILES/tfht/gen/internal/read0289
        echo "echo $SEQFILES/tfht/gen/internal/read0289 RETAINED" >> ${JON}_WK3n
        echo "echo $SEQFILES/tfht/gen/internal/read0289 DELETED" >> ${JON}_WK3a
        echo "rm -f $SEQFILES/tfht/gen/internal/read0289" >> ${JON}_WK3a
        setenv DD_STEP010_INTREAD $SEQFILES/tfht/gen/internal/read0289
        setenv PDD_STEP010_INTREAD $DD_STEP010_INTREAD
        setenv PLN_STEP010_INTREAD $DD_STEP010_INTREAD
        echo "unsetenv PDD_STEP010_INTREAD" >> ${JON}_WK3n
        echo "unsetenv PDD_STEP010_INTREAD" >> ${JON}_WK3a
        echo "unsetenv PLN_STEP010_INTREAD" >> ${JON}_WK3n
        echo "unsetenv PLN_STEP010_INTREAD" >> ${JON}_WK3a
        execute $PROCLIB/TFTDISP3,
msg="AR1=TFTVPRD,AR2=TFTVPRD,PARM1=N13320345009999TFTVPRDTFTVPRD,SYSOUTA=A"
        setenv STATUS `cat status.${JON}`
        unsetenv LRECL_STEP010_INTREAD
        unsetenv DD_STEP010_INTREAD
        if ( $STATUS == 0 ) then
            source ${JON}_WK3n
        else
            source ${JON}_WK3a
            setenv JOBSTATUS 1
        endif
        echo $STEPNAME step was executed cond code $STATUS
        echo 'step $START'
        echo 'step $STEPNAME stop\ \ at TIME=`ebmtime`'
    else
        echo "step $STEPNAME is bypassed"
    endif
endif
unsetenv PARMPROC
#***** end step STEP2 *****

#***** begin step STEP3 *****
STEP3:
#//STEP3   EXEC PGM=IEBGENER
setenv STEPNAME STEP3
setenv JOB_COND `${JON}_JCOND`
echo "_${STEPNAME}_ 0" > ${JON}_RET
if ( $JOB_COND == 1 ) then
    echo "step $STEPNAME bypassed"
    echo "echo 1" > ${JON}_JCOND
else
    if ( $JOBSTATUS == 0 ) then
        setenv START "$STEPNAME start at TIME=`ebmtime`"
        echo "starting STEP -- $STEPNAME"
        setenv S_CKPFIL 0
        cat <$DEVNULL > ${JON}_WK1
        cat <$DEVNULL > ${JON}_WK2a
        cat <$DEVNULL > ${JON}_WK2n
        setenv DD_SYSIN $DEVNULL
        setenv LN_SYSIN $DEVNULL
        echo "echo $SEQFILES/tfht/gen/internal/read0289 RETAINED" >> ${JON}_WK2n
        echo "echo $SEQFILES/tfht/gen/internal/read0289 RETAINED" >> ${JON}_WK2a
        setenv DD_SYSUT1 $SEQFILES/tfht/gen/internal/read0289
        if ( ! -d ${SYSOUTDIR}/${JOBNAME}/${STEPNAME} ) then
            mkdir -p ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}
        endif
        touch ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/INTRDR_${JON}
    endif
endif

```

```

echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/INTRDR_${JON} SYSOUT" >> ${JON}_WK2n
echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/INTRDR_${JON} SYSOUT" >> ${JON}_WK2a
setenv DD_SYSUT2 ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/INTRDR_${JON}
IEBGENER
setenv STATUS `cat status.${JON}`
unsetenv DD_SYSUT2
unsetenv DD_SYSUT1
unsetenv DD_SYSIN
unsetenv LN_SYSIN
if ( $STATUS == 0 ) then
    source ${JON}_WK2n
else
    source ${JON}_WK2a
    setenv JOBSTATUS 1
endif
echo $STEPNAME step was executed cond code $STATUS
echo 'step $START'
echo 'step $STEPNAME stop\ \ at TIME=`ebmtime`'
else
    echo "step $STEPNAME is bypassed"
endif
endif
cat ${JON}_RET >> ${JON}_RETCOD
unsetenv PARM
#***** end step STEP3 *****

#***** begin step STEP4 *****
STEP4:
#//STEP4 EXEC PGM=FCHPGML,PARM='CA7STAT',COND=EVEN,REGION=2048K
setenv STEPNAME STEP4
setenv JOB_COND `${JON}_JCOND`
echo "_${STEPNAME}_ 0" > ${JON}_RET
if ( $JOB_COND == 1 ) then
    echo "step $STEPNAME bypassed"
    echo "echo 1" > ${JON}_JCOND
else
    setenv START "$STEPNAME start at TIME=`ebmtime`"
    echo "starting STEP -- $STEPNAME"
    setenv S_CKPPFILE 0
    cat <$DEVNULL > ${JON}_WK1
    cat <$DEVNULL > ${JON}_WK2a
    cat <$DEVNULL > ${JON}_WK2n
    setenv PARM "'CA7STAT'"
    echo "echo $SEQFILES/fchvp/site/table RETAINED" >> ${JON}_WK2n
    echo "echo $SEQFILES/fchvp/site/table RETAINED" >> ${JON}_WK2a
    setenv DD_FCHSITE $SEQFILES/fchvp/site/table
    echo "echo $SEQFILES/fchvp/control/primary RETAINED" >> ${JON}_WK2n
    echo "echo $SEQFILES/fchvp/control/primary RETAINED" >> ${JON}_WK2a
    setenv DD_FCHCCFP $SEQFILES/fchvp/control/primary
    echo "echo $SEQFILES/fcfch/control/backup RETAINED" >> ${JON}_WK2n
    echo "echo $SEQFILES/fcfch/control/backup RETAINED" >> ${JON}_WK2a
    setenv DD_FCHCCFB $SEQFILES/fcfch/control/backup
    if ( ! -d ${SYSOUTDIR}/${JOBNAME}/${STEPNAME} ) then
        mkdir -p ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}
    endif
    touch ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/STATPRT_${JON}
    echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/STATPRT_${JON} SYSOUT" >> ${JON}_WK2n
    echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/STATPRT_${JON} SYSOUT" >> ${JON}_WK2a
    setenv DD_STATPRT ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/STATPRT_${JON}
    if ( ! -d ${SYSOUTDIR}/${JOBNAME}/${STEPNAME} ) then
        mkdir -p ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}
    endif
    touch ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/PRINT_${JON}
    echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/PRINT_${JON} SYSOUT" >> ${JON}_WK2n
    echo "echo ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/PRINT_${JON} SYSOUT" >> ${JON}_WK2a
    setenv DD_PRINT ${SYSOUTDIR}/${JOBNAME}/${STEPNAME}/PRINT_${JON}
    setenv DD_PRINT2 $DEVNULL
    runpgm fun=FCHPGML
    setenv STATUS $status
    unsetenv DD_PRINT2
    unsetenv DD_PRINT
    unsetenv DD_STATPRT
    unsetenv DD_FCHCCFP
    unsetenv DD_FCHCCFB
    unsetenv DD_FCHSITE
    if ( $STATUS == 0 ) then
        source ${JON}_WK2n
    else
        source ${JON}_WK2a
        setenv JOBSTATUS 1
    endif
endif

```

---

```
                endif
                echo $STEPNAME step was executed cond code $STATUS
                echo 'step $START'
                echo 'step $STEPNAME stop\ \ at TIME=`ebmtime`'
endif
cat ${JON}_RET >> ${JON}_RETCOD
unsetenv PARM
#***** end step STEP4 *****

rm -f ${JON}_WK1
if ( -d ${SYSOUTDIR}/${JOBNAME} ) then
    OW ${SYSOUTDIR}/${JOBNAME}
endif
if ( -d ${FORMS}/${JOBNAME} ) then
    OW ${FORMS}/${JOBNAME}
endif
if ( $JOBSTATUS == 0 ) then
    source ${JON}_WK4n
else
    source ${JON}_WK4a
endif
rm -f ${JON}_WK2a
rm -f ${JON}_WK2n
rm -f ${JON}_WK3a
rm -f ${JON}_WK3n
rm -f ${JON}_WK4a
rm -f ${JON}_WK4n
rm -f ${JON}_RETCOD
rm -f ${JON}_RET
rm -f ${JON}_JCOND
rm -f status.${JON}
echo '$JOBSTART'
echo 'JOB $JOBNAME stop\ \ at TIME=`ebmtime`'
exit $JOBSTATUS
#***** END OF JOB JOB00289 *****
```

---

## Appendix 3

Example UniKix BPE software and Levi, Ray and Shoup VPSX Script Extract

### Example Script Extract

```
#!/bin/ksh
#####
#*
#* Copyright (c) 2008 by Clarity Solution, Inc.
#* All rights reserved.
#*
#####
#
#
# Name:          post_exec_pgm.LRS
# Purpose:      Submit a print request to LRS's VPSX product
#               using the VPSX lrsq command for adding entries to the
#               print queue.
#
#               If using the LRS VPSX product,
#               $PUBLIC/bin/post_exec_pgm must point to this script.
#
#               This script is invoked at the end of each job step to determine
#               if there are sysout datasets to be printed. For each of the
#               SYSOUT datasets generated in the current step, it formats and
#               executes the LRS's VPSX lrsq command based on environment
#               variables set in the batch job script
#
#               The format of the VPSX print request command is:
#
#               lrsq /[option][=:] [value]
#
# where:
#
# /File=name    usage      : Specifies the full pathname of the
#                   file(s) to be printed
# /Queue=name   usage      : Specifies the name of the VPSX printer that
#                   should receive this output
# /Copies=<n>   usage      : Specifies the number of copies to be printed.
#                   default   : 1
#                   JCL source : The DD or OUTPUT COPIES parameter
#
# /Hold=Y|N    usage      : Sets print file status to Held; it will not
#                   : be processed until manually released
#                   default   : N - no hold
#                   JCL source : DD HOLD parameter
#
# /FORMDEF=form usage      : Specifies the paper type, or forms to be used
#                   default : standard printer default
#                   JCL source: SYSOUT DD parameter or OUTPUT FORMS parameter
#
# /Prty=<1-255> usage      : Specifies the priority of the print request.
#                   default  : 10
#                   JCL source : OUTPUT PRTY parameter
#
#               Note: VPSX supports a priority range of
#                   1 through 255, with 255 representing the highest
#                   priority. MVS/JCL supports 0 through 255. If
#                   the OUTPUT PRTY parameter specifies a value
#                   greater than 255, or 0, the priority is set to 10.
#
# The lrsq command includes other options that are not strictly related to
# MVS/JCL. This script may be modified to include these options. Refer to
# the VPSX User's Guide for a complete description of the lrsq command.
#####
```

**Learn More**

For more information on mainframe migration solutions from Clernity and associated products please visit:

[www.clerity.com](http://www.clerity.com)



9930 Derby Lane, Suite 202 • Westchester, IL 60154  
Phone 1-888-2-REHOST (or 1-630-981-6100)

© 2010 Clernity Solutions, Inc. All rights reserved. Clernity and UniKix are trademarks, or registered trademarks, of Clernity Solutions, Inc. in the United States and other countries.

IBM, Tivoli, z/OS, and z/VSE are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. All other marks are the property of their respective owners.