z/TPF Data Event (DEVT) Driver

CONTENTS
_____

1.0 Prerequisites
_____

The release levels are the following:
        o z/TPF (PUT 11 or later) with APAR PJ42834 installed
        o z/TPFDF (PUT 11 or later) with APAR PI32332 installed

The build tool is the following:
        o maketpf build tools

2.0 Introduction
_____

The z/TPF data event starter kit shows how the z/TPF system automatically creates data events when an application updates a z/TPFDF database and how an event consumer on a remote system receives and processes those data events.

For this starter kit, the z/TPF data event (DEVT) driver represents the application that updates a sample z/TPFDF Credit database. This document describes how to build and install the DEVT driver on your z/TPF system.

The DFDLtoXMLSample Java application represents the event consumer that processes data events received from the z/TPF system. For more information on installing the DFDLtoXMLSample application, see the z/TPF Data Event Starter Kit User's Guide (DEVTStarterKit_UsrGde.pdf).

3.0 Installation
_____

1) Use FTP to transfer the tar file (DEVTDriver.tar.gz) to your Linux system. This
   can be placed in any directory as a holding location, for example, /tmp/ztpftar

2) Create a root directory that will hold the unpacked files, for
   example, /ztpfdrvs

3) Extract the source code from the tar file by entering the
   following commands:
   cd /ztpfdrvs
   tar -xvzf /tmp/ztpftar/DEVTDriver.tar.gz

   The driver source files will be extracted in the following directory
   structure:
       devt/app_drvs_sys.mak
       devt/c_bioprf.h
       devt/c_id11ed.h
       devt/c_dr12ed.h
       devt/c_dr14ed.h
       devt/c_prfusr.h
       devt/credit_trxns_usrctx.h
       devt/devt.cntl
       devt/devt.loadfile
       devt/dr11ed.mac
       devt/dr12ed.mac
       devt/dr14ed.mac
       devt/qdva.cpp
       devt/qdva.mak
       devt/qdvc.h
       devt/qdvc.mak
       devt/qdvc_addcard.cpp
       devt/qdvc_addcharge.cpp
       devt/qdvc_addsec.cpp
       devt/qdvc_adduser.cpp
       devt/qdvc_build.cpp
       devt/qdvc_delcard.cpp

```
devt/qdvc_delcharge.cpp
devt/qdvc_delsec.cpp
devt/qdvc_deluser.cpp
devt/qdvc_dispcard.cpp
devt/qdvc_dispcharge.cpp
devt/qdvc_dispec.cpp
devt/qdvc_dispuser.cpp
devt/qdvc_modcard.cpp
devt/qdvc_moduser.cpp
devt/qdvp.cpp
devt/qdvp.h
devt/qdvp.mak
devt/qdvp_grammar.h
devt/qdvp_parms.cpp
devt/qdvt.cpp
devt/qdvt.mak
devt/qdvz.cpp
devt/qdvz.h
devt/qdvz.mak
devt/qdvz_help.cpp
devt/uf92.asm
devt/eventspecs/Credit_Trxns.de.evspec.xml
devt/eventspecs/IPSendBINAdapter.evda.xml
devt/schemas/Credit_Trxns.de.dfdl.xsd
devt/schemas/Credit_Trxns_Usrctx.user.dfdl.xsd
devt/schemas/DR12ED.tpfdf.dfdl.xsd
```

4) Create a maketpf.cfg file with the following contents:

```
APPL_ROOT := /ztpfdrvs
TPF_ROOT := /ztpf
TPF_BSS_NAME := BSS
#TPF_SS_NAME :=
#USER_VERSION_CODE :=
```

   a) Set APPL_ROOT to the directory that contains the driver source code
      that was extracted.
   b) Set TPF_ROOT to the directory that contains the z/TPF source code.
   c) Set TPF_BSS_NAME to the basic subsystem name of your z/TPF system. By
      default, this is set to BSS.
   d) Optionally, set the TPF_SS_NAME to the subsystem name.
   e) Optionally, set USER_VERSION_CODE to any desired two-character string.
      The two-character string set will be appended to the shared objects built.
      By default, this is set to null.

   For details about these variables, see the maketpf tool documentation.

5) Add the #CREDIDX fixed record type to your SIP Stage 1 deck, build
   the FACE Table (FCTB), and load the FCTB to your z/TPF system.

   a) In your SIP Stage 1 deck, use the RAMFIL macro to define a fixed file
      record with record ID #CREDIDX, a minimum of 880 ordinals, and a size

of 4K. This fixed record is used by the DEVT driver as the index
record for the credit database.

   b) Using the bldtpf command, build the FACE Table (FCTB) for the SIP Stage
      1 deck modified in step 5a. For example,
         bldtpf -fctb /ztpf/bss/src/sip.asm

   c) Load the FCTB built in step 5b to your z/TPF system. If you are using
      the image loader (TLDR) to load the FCTB, you can load the FCTB and
      IPAT (step 6d below) in the same load deck.

6) Build the USRSTUB program and IPAT after adding the DEVT driver control file
   to your user control file.

   a) Add the following line to your user control file base/cntl/usr.cntl
         include devt/devt.cntl

   b) Build the USRSTUB program to generate stubs for all user programs
      using the following command:
         maketpf USRSTUB -f

   c) Rebuild IPAT to incorporate the changes you made in usr.cntl
         maketpf ipat -f

   d) Load the IPAT built in step 6c to your z/TPF system.

7) Run the maketpf tool with the accompanied control file (devt.cntl) to
   assemble, compile, and link the driver programs:
         bldtpf /ztpfdrvs/devt/devt.cntl

8) Modify the business event dispatch adapter for the DEVT driver. The
   dispatch adapter is defined in file devt/eventspecs/IPSendBINAdapter.evda.xml
   and uses TCP/IP to send the event message data events to the remote system.
   Follow the instructions in the dispatch adapter to set the IP address and
   port number to the IP address and port of remote system where the data event
   starter kit will be running.

9) Use the standard load procedure to transfer and load the DEVT driver
   programs, business event specifications, and DFDL schemas to your
   z/TPF system.
         loadtpf devt/devt.cntl
         loadtpf devt/devt.loadfile

For more information about building and loading programs to the z/TPF system,
see the Program Management topic in the z/TPF product documentation.

4.0 Customization
_____

1) Update base/rt/cvzz.asm (or the tool that executes driver programs) to make
   an entry for the DEVT driver. The shared object QDVP is the main entry point

for this driver.

2) Add the DBDEF program UF92 to your local_mod version of tpfdf/macro/dfuex.mac.
   This change will be incorporated when UFC8 is rebuilt in the next step.
   For example:
        GETPC NAME=UF92,LOCK=YES,ADDR=R2   LOCK PROGRAM IN CORE
        ENTRC UF92   Devt DRIVER DBDEFS
        BR   R7

3) Build and load the updated CVZZ and UFC8 programs to the z/TPF system.

4) Enter ZUDFM DEF INIT to rebuild the DBDEF index table.

5.0   Initialize the Database
───────────────────────────────────

Enter the following commands to initialize the credit database and create
the predefined data in the database.

1)   ZUDFM INIT B211 (enter twice)
2)   ZTEST DEVT CREDIT BUILD

6.0   Configure and Enable Data Events
──

1) Set up and configure z/TPF business events for your z/TPF system.
   For information about how to configure and enable business event
   processing, see "Configuring and enabling business event processing" in the
   PUT 11 (or later) z/TPF product documentation.

2) The DEVT driver uses the system default dispatch queue(i.e.
IBEV.UNORDERED.DISPATCH.QUEUE)
   as the dispatch queue for the Credit_Trxns data event. Define and enable this queue
   if not done in step 1. To monitor this queue, the following command can be used
      ZMQSC DISPLAY QL-IBEV.UNORDERED.DISPATCH.QUEUE STAT

3) If not done as part of step 1 above, use the following command to enable
   data events on your z/TPF system.
      ZBEVF ENABLE DATA

7.0   Define and Deploy Data Events
──────────────────────────────────────

1) To define and deploy data events for a z/TPFDF file, you have to create the
   necessary components (business event specification, dispatch adapters, and
   custom programs) for each z/TPFDF file that you want to enable for data
   events. This driver already includes all of the components needed to define
   an event message data event called "Credit_Trxns". Credit_Trxns is an event
   message data event for the z/TPFDF file DR12ED, which contains information on

authorized users and credit card transactions.

The components used to define and deploy the event message data event
included with this driver are listed below. All build instructions and
customizations related to these components are covered in previous steps
and no further action is required in this step; the remainder of this step
is for informational purposes only.

For information about how you would build these components yourself, see
"Defining and deploying business events" in the PUT 11 (or later) z/TPF
product documentation.

-Business event specification for the Credit_Trxns event message data event
    devt/eventspecs/Credit_Trxns.de.evspec.xml

-Dispatch adapter to send binary event messages over TCP/IP
    devt/eventspecs/IPSendBINAdapter.evda.xml

-DFDL schema file that defines the format of the event message. This is the
 "top-level" schema that references the other DFDL schema files used to
 define the event message.
    devt/schemas/Credit_Trxns.de.dfdl.xsd

-DFDL schema file for the optional event user context area
    devt/schemas/Credit_Trxns_Usrctx.user.dfdl.xsd

-DFDL schema file for the DR12ED z/TPFDF file that defines the format of
 the DR12ED LRECs
    devt/schemas/DR12ED.tpfdf.dfdl.xsd

-Application enrichment program
    devt/qdva.cpp
    devt/qdva.mak

-Custom transmit program to send the event message over a TCP/IP connection
    devt/qdvt.cpp
    devt/qdvt.mak

-Update to the DBDEF to enable data events for the DR12ED z/TPFDF file.
 The DEVCLCT and DEVNAME were added to the DBDEF macro for the DR12ED file.
    devt/uf92.asm

2) Use the following command to verify the IPSendBINAdapter is deployed
      ZMDES DISPLAY FILE-IPSendBINAdapter.evda.xml

3) If the IPSendBINAdapter is not deployed, issue the following command to
   deploy the IPSendBINAdapter dispatch adapter.
      ZMDES DEPLOY FILE-IPSendBINAdapter.evda.xml

4) Use the following command to deploy the business event specification
   for the Credit_Trxns data event.
      ZMDES DEPLOY FILE-Credit_Trxns.de.evspec.xml

8.0 Running the driver
_____

Enter ZTEST DEVT CREDIT commands to run the credit database sample driver and
generate data event records. For details on the Credit database layout and the
DEVT driver command syntax, see the z/TPF Data Event (DEVT) Driver User's Guide
(DEVTDriver_UsrGde.pdf).

9.0 NOTICES
_____

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this
information in other countries. Consult your local IBM representative for
information on the products and services currently available in your area.
Any reference to an IBM product, program, or service is not intended to
state or imply that only that IBM product, program, or service may be used.
Any functionally equivalent product, program, or service that does not
infringe any IBM intellectual property right may be used instead. However,
it is the user's responsibility to evaluate and verify the operation of
any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter
described in this information. The furnishing of this information does not
grant you any license to these patents. You can send license inquiries,
in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information,
contact the IBM Intellectual Property Department in your country or send
inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other
country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS INFORMATION
"AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 31BA/Building 008
Mail Drop P369
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

9.1 Trademarks

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

9.2 Warranty

This package is provided on an "as is" basis. There are no warranties, express or implied, including the implied warranties of merchantability and fitness for a particular purpose. IBM has no obligation to provide service, defect correction, or any maintenance for the package. IBM has no obligation to supply any updates or enhancements for the package to you even if such are or later become available.