z/TPF support for MongoDB starter kit readme

NOTE: Before using this information and the product it supports, read the general information under "Notices" in this document.

Contents

This file includes the following information:

1.0 Introduction

The z/TPF support for MongoDB starter kit provides instructions, sample database definitions, and driver programs so that you can explore the z/TPF support for MongoDB.

Use the instructions in this file to set up a sample z/TPFDF database for a remote Java application to access data by using the standard MongoDB query language.

For more information about z/TPF support for MongoDB, see the z/TPF product documentation in IBM Knowledge Center.

2.0 Change history

2015Dec18 Initial version

3.0 Prerequistes

The following list provides the required release levels:
- o z/TPF PUT 12 or later, with APAR PJ42292 and PJ43302 installed
- o z/TPFDF PUT 12 or later, with APAR PI33010 and PI40719 installed

Note: z/TPF support for MongoDB must be built with one of the following compilers:
- o GCC 4.6 version tpf-11r1-13 or later
- o Dignus Systems/C and Systems/C++ 1.98 version 1.98.68 or later

The following build tool is required:
- o maketpf utility

4.0 Installing the starter kit

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

2) Create a root directory to hold the uncompressed files, for example, /ztpf/mongodb/starterkit

3) Extract the source code from the tar file by entering the following commands:

```
cd /ztpf/mongodb/starterkit
tar -xvzf /tmp/ztpftar/mongoDBStarterKit.tar.gz
```

   The following source files will be extracted to the following directory structure:
   - o DBDEFs and DSECTs for the sample z/TPFDF database (z/TPF collections for MongoDB)
     - o temp/uf84.asm // DBDEFs for the sample databases
     - o temp/uf84.mak
     - o temp/dr21bi.mac // DSECT for the Flight collection
     - o temp/dr22bi.mac // DSECT for the Seat collection
     - o temp/dr23.bi.mac // DSECT for the Passenger Name temp/collection
     - o temp/dr25bi.mac // DSECT for the Passenger Number collection
     - o temp/dr26bi.mac // DSECT for the PNR collection

   - o Customized collection descriptors and DFDL schema files for the sample databases.

```
       Collection descriptor:
       o temp/Flight.adbi.xml           // For Flight collection
       o temp/Seat.adbi.xml             // For Seat collection
       o temp/Name.adbi.xml             // For Name collection
       o temp/Number.adbi.xml           // For Number collection
       o temp/PNR.adbi.xml              // For PNR collection

       DFDL schema files:
       o temp/Flight.tpfdf.adbi.xsd     // For Flight collection
       o temp/Seat.tpfdf.dfdl.xsd       // For Seat collection
       o temp/Name.tpfdf.dfdl.xsd       // For Name collection
       o temp/Number.tpfdf.dfdl.xsd     // For Number collection
       o temp/PNR.tpfdf.dfdl.xsd        // For PNR collection

     o Sample TLDR and OLDR load decks to load the FACE table (FCTB), program
attribute
       table (IPAT), database definitions (DBDEFs), collection descriptors, and the
DFDL
       schema files.
       o build/maktpf.cfg
       o build/mongodbSample.tldr
       o build/mongodbSample.oldr

     o Sample MongoDB client application program
       o java/skitBuild.jar     // To populate the sample database
       o java/skitBuild.java    // Source code of skitBuild
       o java/skitTraffic.jar   // To generate transaction to create, update, query,

                                   delete documents for sample database
       o java/skitTraffic.java  // Source code of skitTraffic

  4) Update maketpf.cfg to reference the starter kit directory.
     o Set the APPL_ROOT variable to the directory that contains the driver source
code
       that was extracted.

     o Set the TPF_ROOT variable to the directory that contains the z/TPF source
code.

     o Set the TPF_BSS_NAME variable to the basic subsystem name of your z/TPF
system.
       By default, this variable is set to BSS.

     o Optional: Set the TPF_SS_NAME variable to the subsystem name.

     o Optional: Set the USER_VERSION_CODE variable to any 2-character string.
       The 2-character string is appended to the shared objects that are built.
       By default, this value is set to NULL.

     For more information about these variables, enter man maketpf on your Linux
```

build
system.

5) Define fixed file records for the sample z/TPF collections for MongoDB (z/TPFDF

files).

a. Add the following record type to your SIP stage 1 input deck (RAMFIL statements)

- o For the Flight collection:
  - RECID: #AFLIGHT
  - Size: 4 KB
  - Number of records: 367
- o For the Passenger Name collection:
  - RECID: #APGRNAM
  - Size: 4 KB
  - Number of records: 676
- o For the Passenger Number collection:
  - RECID: #APRGNUM
  - Size: 4 KB
  - Number of records: 499
- o For z/TPFDF automatic indexing:
  - RECID: #APAINFO
  - Size: 4 KB
  - Number of records: 101

b. Build the FACE table (FCTB) to include the fixed file records for the sample z/TPF collections for MongoDB. For example, enter the following command from the build directory:

```
bldtpf -fctb /ztpf/bss/src/sip.asm
```

c. Load and activate the FCTB to your z/TPF system. If you use the image loader (TLDR) to load the FCTB, you can load the FCTB and the IPAT (see step 6) in the same load deck.

6) Add UF84 to the user control file and build the USRSTUB program and IPAT.

a. Build the USRSTUB program to generate stubs for all user programs by entering the following command from the build directory:

```
maketpf USRSTUB -f
```

b. Rebuild the IPAT to incorporate the changes that you made in the usr.cntl file:

```
          maketpf ipat -f
```

c. Load and activate the IPAT to your z/TPF system.

5.0 Customizing the starter kit

Define the sample z/TPF collections for MongoDB (z/TPFDF files) to the z/TPF system:

1) Add the DBDEF program UF84 to your dfuex.mac file. This change is incorporated when UFC8 is rebuilt in the next step. For example:

```
       GETPC NAME=UF84,LOCK=YES,ADDR=R2    LOCK PROGRAM IN CORE
       ENTRC UF84                          z/TPF support for MongoDB starter kit
DBDEFS
       BR    R7
```

2) Build program UFC8:

```
       maketpf ufc8 -f
```

3) Build program UF84:

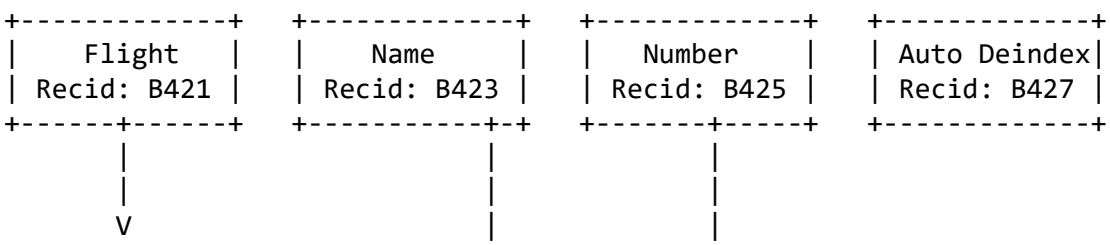
```
       maketpf uf84 temp/uf84.mak -f
```

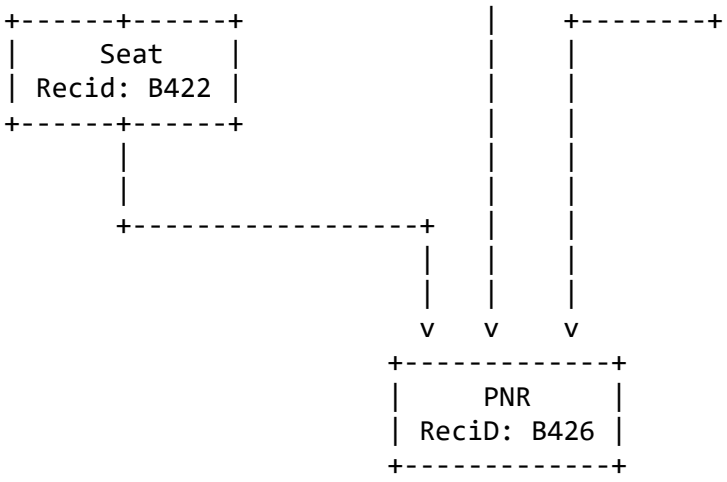
4) Load and activate UFC8 and UF84 to the z/TPF system.

5) Load and activate all collection description files and DFDL schema files to the z/TPF system.
You can use TLDR or OLDR to load the files. Sample load decks are available in the
build directory.

6.0 Initializing the database

The sample Flight database consists of 5 collections: the Passenger Name, Passenger Number,
and Flight collections are index files, the Seat collection is a second level index file,
and the PNR collection is the detail file.





Enter the following commands to initialize the sample databases:

1) ZUDFM DEF INIT
2) ZUDFM INIT B421 (enter this command twice)
3) ZUDFM INIT B423 (enter this command twice)
4) ZUDFM INIT B425 (enter this command twice)
5) ZUDFM INIT B427 (enter this command twice)

7.0 Defining the z/TPF server for MongoDB

Complete the following steps to define and start the z/TPF server for MongoDB:

1) Define the server:

   zinet add s-mongo model-daemon pgm-cads xparm---noauth

   Notes:
   o Port 27017 is the default port number
   o Use --noauth to bypass authentication for the starter kit

2) Start the server:

   zinet start s-mongo

8.0 Populating the database

Run the skitBuild driver from your Linux system as a MongoDB Java client to populate the sample database.

For example, to build 300 PNR records in the sample database, enter the following command

from the java directory:

  java -jar skitBuild.jar n-300 ip-<tpf_host_ip_address>

  where <tpf_host_ip_address> is the z/TPF host IP address where the z/TPF server
for
  MongoDB resides.

For more information about the options available with the skitBuild driver, enter
the
following help command:

  java -jar skitBuild.jar

9.0 Updating the database

Run the traffic driver from your Linux system as a MongoDB Java client to access the
sample database:

  java -jar skitTraffic.jar n-300 ip-<tpf_host_ip_address>

  where <tpf_host_ip_address> is the z/TPF host IP address where the z/TPF server
for
  MongoDB resides.

For more information about the options available with the skitTraffic driver, enter
the
following help command:

  java -jar skitTraffic.jar

10.0 Other methods to access data with z/TPF support for MongoDB

In addition to writing a MongoDB application to access data on the z/TPF system, you
can
access data on z/TPF by using a MongoDB client shell.

  o Download and install a MongoDB client shell from the MongoDB website
  o Download and install a third party vendor tool, such as Robomongo, from the
vendor's
    website.

11.0 Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document 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 document. The furnishing of this document 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, MD-NC119
Armonk, NY 10504-1785
US

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "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 jurisdictions 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 publication. IBM may make improvements and/or changes in the product(s)
and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only
and do not in any manner serve as an endorsement of those websites. The materials at

those websites are not part of the materials for this IBM product and use of those websites
is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

11.1 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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries,
or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of
Oracle and/or its affiliates.

11.2 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.