

OpenFrame OSC Region Build Guide



Update History

Date	Written by	Comments
AUG 29, 2019	GPS1 Kelsey Lee, Nick Ahn	First Created

* Check the resources of the region from SD Macro file.

1) Check which tranclasses are used for each region.

```
DEFINE TRANCLASS(transaction_id)
  GROUP(resourcegroup_name)
  MAXACTIVE(number)
  PURGETHRESH(NO | number)
```

2) Check which group lists are used for each region.

```
ADD GROUP(resourcegroup_name01) LIST(grouplist)
ADD GROUP(resourcegroup_name02) LIST(grouplist)
ADD GROUP(resourcegroup_name03) LIST(grouplist)
```

* BUILD the region and configure it in TMAX configuration.

1) Build the region with oscbuild tool. (options can vary.)

```
oscbuild -o LINUX64 -d TIBERO -s region_name -b OFCOBOL
```

```
oscbuild version 7.0.3(11) obuild@tplinux64:ofs7/osc(#1) 2017-11-29 20:51:10
OpenFrame OSC Server Build Tool
Usage : oscbuild -o <OS> [-d <DB>] [-s <server>] [-f <source_file>] [-i] [-p]
        [-g] [-c '<option_string>'] [-l '<option_string>']
        [-b <cobol-compiler>]
        | oscbuild [-h | -v]
Options:
  -o AIX32 | AIX64 | Select OS type
      HP32 | HP64 |
      SUN32 | SUN64 |
      LINUX32 | LINUX64
  -d ORACLE | Select RDB type
      TIBERO |
      DB2_32 | DB2_64 | DB2_64_STAT
  -s <server> Set osc server binary name (maximum 8 bytes)
  -f <source_file> Specify source file for osc server
  -i Set for using EXEC DLI commands
  -c '<option_string>' Set additional options for compile time
  -l '<option_string>' Set additional options for linking time
  -p Set for AIX PL/I support
  -g Produce server debugging information
  -b MFCOBOL | OFCOBOL Select COBOL Compiler
  -h Display this information
  -v Display version information
```

2) Deploy the region server module to APPDIR directory. (from oframe.m)

```
cp -a region_name $TMAXDIR/appbin/
APPDIR = "/opt/tmaxapp/OpenFrame/core/appbin"
```

3) Set the tmax configuration for the region server. (\$TMAXDIR/config)

Each option should be configured. (Refer [3.2.4. SERVER](#) from [Tmax_5_SP2_Fix#1_Administrator_s_Guide_v2.1.5.1_en](#))

For the specific setting, it needs to be discussed with the customer.

Use the server setting from the default region(OSCOIVP1). (copy -> rename -> edit)

[example]

```

region_name          SVGNAME = svgbiz,
                      MIN = 3,
                      MAX = 10,
                      SCHEDULE = FA,
                      CLOPT = "-n -o $(SVR)_$(CDATE).out -e $(SVR)_$(CDATE).err"
region_nameC          SVGNAME = svgbiz,
                      TARGET = region_name,
                      CONV = 0,
                      MIN = 1,
                      MAX = 128,
                      SCHEDULE = FA,
                      CLOPT = "-n -o $(SVR)_$(CDATE).out -e $(SVR)_$(CDATE).err"
region_nameOMC        SVGNAME = svgbiz,
                      TARGET = oscossvr,
                      MIN = 1,
                      MAX = 5,
                      SCHEDULE = FA,
                      CLOPT = "-o $(SVR)_$(CDATE).out -e $(SVR)_$(CDATE).err -x
OSCOSSVRSVC1:region_name_OMC1,OSCOSSVRSVC2:region_name_OMC2,OSCOSSVRMON:region_name
_MON,OSCOSSVR_ST:region_name_ST" <- This should match the one from default region.
region_nameTL          SVGNAME = svgbiz, MAX = 1, SVRTYPE = UCS, target=osctlsvr,
                      CLOPT = "-o $(SVR)_$(CDATE).out -e $(SVR)_$(CDATE).err -x
                      OSCTLSVRSVC:region_name_TL"
region_name_tranclass SVGNAME= svgbiz,
                      TARGET = region_name
                      MIN = 1,
                      CLOPT = "-n -o $(SVR)$ (CDATE).out -e $(SVR)$ (CDATE).err"

```

```

region_name          SVRNAME = region_name
region_nameP          SVRNAME = region_name
region_nameC          SVRNAME = region_nameC
region_nameM          SVRNAME = region_nameC
region_name_TL        SVRNAME = region_nameTL
region_name_OMC1      SVRNAME = region_nameOMC
region_name_OMC2      SVRNAME = region_nameOMC
region_name_MON       SVRNAME = region_nameOMC
region_name_ST        SVRNAME = region_nameOMC
region_name_tranclass SVRNAME = region_name_tranclass

```

4) Compile the tmax configuration file.

```
cfl -i oframe.m
```

*** Generate online system VSAM files (TDQ,TSQ,SD) for the region.**

1) Generate SD dataset.

```
idcams define -t CL -n SD dataset name -o KS -k 18,0 -b 32768 -l 128,32760 -s
1024,128,128 -v DEFVOL
```

2) Generate TDQ dataset.

```
idcams define -t CL -n TDQ dataset name -o KS -k 8,0 -l 128,32760 -b 32767 -s
1024,128,128 -v DEFVOL
```

3) Generate TSQ(KEY and DATA) dataset.

```
idcams define -t CL -n TSQ KEY dataset name -o KS -k 16,0 -l 64,64 -s 1024,128,128
-v DEFVOL

idcams define -t CL -n TSQ DATA dataset name -o KS -k 18,0 -l 128,32760 -b 32767 -
s 1024,128,128 -v DEFVOL
```

4) Register the CSD resource definition to SD dataset.

Register system online resource

```
oscsdgen -c -d SD dataset name $OPENFRAME_HOME/osc/resource/osc.dat
```

Register User CSD

```
oscsdgen -c -d SD dataset name user resource file
```

```
oscsdgen version 7.0.3(10) obuild@tplinux64:ofs7/osc(#1) 2017-11-29 20:51:10
OSC System Definition(OSCSD) Update Utility
Usage: oscsdgen -c {-r <region>|-d <dataset>} <file>
      | oscsdgen -f {-r <region>|-d <dataset>}
<group_name>:<resource_name>:<resource_type>
      | oscsdgen [options]
<file>          Specify input file
-c              Generate SD
-f              Delete SD resource specified
-d <dataset>    Specify OSCSD dataset name
-r <region>     Specify OSC region name
<group_name>    SD GROUP name of specified resource
<resource_name> SD resource name
<resource_type> SD type of specified resource

{connection|file|journalmodel|pipeline|program|tdq|terminal|transaction|tsmodel|ty
peterm|webservice|tranclass|

enqmodel|lsrpool|mapset|partitionset|profile|sessions|tcpipSERVICE|urimap|library}
Options:
-h              Display this information
-v              Display version information
```

*** Create and edit Online region configuration files. (copy -> rename -> edit)**

1) Copy the default region(OSCOIVP1) configuration files and rename it.

\$OPENFRAME_HOME/config/

osc.*region_name*.conf & osc.*region_name*TL.conf

```
cp osc.OSCOIVP1.conf osc.region_name.conf
cp osc.OSCOIVP1TL.conf osc.region_nameTL.conf
```

2) Edit the region configuration file.

***** You need to make sure the shared memory key and port setting are different for each region *****

osc.*region_name*.conf

```
[GENERAL]
JOBID=STD00001
SYS_SHMKEY=70454           <- system shared memory key
USR_SHMKEY=70554           <- user shared memory key
TC_PATH=$OPENFRAME_HOME/temp/region_name_TC
MAPDIR=$OPENFRAME_HOME/osc/region/region_name/map <- check $OPENFRAME_HOME path
TBLDIR=$OPENFRAME_HOME/osc/region/region_name/tbl <- check $OPENFRAME_HOME path
TDLDIR=$OPENFRAME_HOME/osc/region/region_name/tcl <- check $OPENFRAME_HOME path

[SD]
DSNAME=SD dataset name
GRPLIST=BASELIST,CICBLIST

[TDQ]
TDQ_INTRA_DSNAME=TDQ dataset name
TDQ_LOG_ADDRESS=127.0.0.1:5954 <- TDQ log address port

[TSQ]
QDATA_DSNAME=TSQ DATA dataset name
QINFO_DSNAME=TSQ KEY dataset name
TSQ_SHMKEY=70654

[TRANCLASS]
DEFAULT_TRANCLASS=DFHTCL00 <- default tranclass
                        * check the default tranclass for the region.
```

osc.*region_name*TL.conf

```
[OSCTLSVR]
PORT=5954 <- TL server port (Use TDQ_LOG_ADDRESS port)
```

* Create MAPDIR, TBLDIR, TDLDIR directory for the region.

You can copy the default MAPDIR, TBLDIR, TDLDIR directory.

```
cp -a $OPENFRAME_HOME/osc/region/OSCOIVP1 $OPENFRAME_HOME/osc/region/region_name
```

* Modify the TDL configuration file for the region.

\$OPENFRAME_HOME/osc/region/*region_name*/tdl/config/tdl.cfg

```
# shared memory key
SHMKEY=70740                                <- It should be different from other regions.
```

* Create the region memory.

Use osctdlinit tool.

```
osctdlinit region_name
```

* Add the region to the region list for booting the region up when oscboot.

\$OPENFRAME_HOME/config/osc.region.list

```
region_name
```

* Reboot Openframe to make the region server setting effective and start the region server.

```
oscdwn          <- OpenFrame server down
oscboot         <- OpenFrame server boot
```