

# Maintaining A Separate Test Environment

This section describes a method to set up a temporary test copy of phases updated by a program fix. The method described is intended as an example. Its relevance depends on the installation standards you use for library maintenance.

The example scenario uses MSHP in a single z/VSE machine to control both the standard production Adabas library and an additional testing library or sublibrary used to validate recently applied program fixes.

After restoring the standard Adabas library and defining it to MSHP, an additional test library or sublibrary can be defined.

Object modules can then be copied from the standard library as required, and controlled with MSHP using a different z/VSE system history file. Using the same component ID as for the standard environment (9001-ADA-00-*vrs*) ensures that the ZAP source remains common to both environments.

The test version of a phase is then invoked by placing the test library or sublibrary at the head of the LIBDEF PHASE search chain.

The setup jobs required to implement this environment are described in detail below. Note that the first three steps form part of the standard installation process.

## to setup the separate test environment:

1. Define standard Adabas library.

For a sample job, see the section *Installing the Adabas Release Tape*.

2. Restore standard Adabas library.

For a sample job, see the section *Installing the Adabas Release Tape*.

3. Define standard Adabas to MSHP.

### **Note:**

This job uses the history file identified by the IJSYSHF label in the z/VSE standard label area.

```
// EXEC MSHP
ARCHIVE ADAvrs
COMPRISES 9001-ADA-00
RESOLVES 'SOFTWARE AG - ADABAS.ADAvrs'
ARCHIVE 9001-ADA-00-vrs
RESIDENCE PRODUCT=ADAvrs -
PRODUCTION=SAGLIB.adannn -
GENERATION=SAGLIB.adannn
/*
```

where *vrs* is the Adabas version, revision, and system maintenance (SM) level and *adannn* is the sublibrary name for standard Adabas.

4. Create test sublibrary and copy object modules to it.

```
// DLBL SAGLIB,'adabas.adannn.library'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=volser,SHR
// EXEC LIBR
DEFINE SUBLIB=SAGLIB.adatst
CONNECT SAGLIB.adannn:SAGLIB.adatst
COPY *.OBJ LIST=Y REPLACE=Y
/*
```

where *adabas.adannn.library* is the physical name of the standard Adabas library, *volser* is the volume on which library resides, *adannn* is the sublibrary name for standard Adabas, and *adatst* is the sublibrary name for testing Adabas.

5. Create additional system history file for test environment and define test Adabas to it.

```
// ASSGN SYS020,DISK,VOL=volhis,SHR
// EXEC MSHP
CREATE HISTORY SYSTEM
DEFINE HISTORY SYSTEM EXTENT=start:numtrks -
UNIT=SYS020 -
ID='sag.test.system.history.file'
ARCHIVE ADAvrs
COMPRISES 9001-ADA-00
RESOLVES 'SOFTWARE AG - ADABAS Vvrs'
ARCHIVE 9001-ADA-00-vrs
RESIDENCE PRODUCT=ADAvrs -
PRODUCTION=SAGLIB.adatst -
GENERATION=SAGLIB.adatst
/*
```

where *volhis* is the volume on which test system history file resides, *start* is the start of extent on which test system history file resides, *numtrks* is the length of extent on which test system history file resides, *sag.test.system.history.file* is the physical name of test system history file, *vrs* is the Adabas version, and *adatst* is the sublibrary name for testing Adabas.

6. Apply zap to test environment.

```
// DLBL IJSYSHF,'sag.test.system.history.file'
// EXTENT SYS020,,,start,numtrks
// ASSGN SYS020,DISK,VOL=volhis,SHR
// DLBL SAGLIB,'adabas.adannn.library'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=volser,SHR
// EXEC MSHP
CORRECT 9001-ADA-00-vrs : ADnnnnn
AFFECTS MODULE=modname
ALTER offset hexold : hexnew
INVOLVES LINK=lnkname
/*
```

where *sag.test.system.history.file* is the physical name of test system history file, *start* is the start of extent on which test system history file resides, *numtrks* is the length of extent on which test system history file resides, *volhis* is the volume on which test system history file resides, *adabas.adannn.library* is the physical name of the standard Adabas library, *volser* is the volume on which library resides, *vrs* is the Adabas version, *nnnnn* is the Adabas fix number, *modname* is the Adabas object module to be zapped and then relinked, *offset* is the hexadecimal offset to the beginning of the zap, *hexold* is the verify data for the zap, *hexnew* is the replace data for the zap, and *lnkname* is the link book for the phase affected.

## 7. Invoke updated test phase.

```
// DLBL SAGLIB,'adabas.adannn.library'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=volser,SHR
// LIBDEF PHASE,SEARCH=(SAGLIB.adatst,SAGLIB.adannn,...)
...
```

where *adabas.adannn.library* is the physical name of the standard Adabas library, *volser* is the volume on which library resides, *adatst* is the sublibrary name for testing Adabas, and *adannn* is the sublibrary name for standard Adabas.

## 8. Apply zap to standard environment.

**Note:**

This job uses the history file identified by the IJSYSHF label in the z/VSE standard label area.

```
// DLBL SAGLIB,'adabas.adannn.library'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=volser,SHR
// EXEC MSHP
CORRECT 9001-ADA-00-vrs : ADnnnnn
AFFECTS MODULE=modname
ALTER offset hexold : hexnew
INVOLVES LINK=lnkname
/*
```

where *adabas.adannn.library* is the physical name of the standard Adabas library, *volser* is the volume on which library resides, *vrs* is the Adabas *version*, *nnnnn* is the Adabas fix number, *modname* is the Adabas object module to be zapped and then relinked, *offset* is the hexadecimal offset to the beginning of the zap, *hexold* is the verify data for the zap, *hexnew* is the replace data for the zap, and *lnkname* is the link book for the phase affected.

## 9. Invoke standard phase.

```
// DLBL SAGLIB,'adabas.adannn.library'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=volser,SHR
// LIBDEF PHASE,SEARCH=(SAGLIB.adannn,...)
...
```

where *adabas.adannn.library* is the physical name of the standard Adabas library, *volser* is the volume on which library resides, and *adannn* is the sublibrary name for standard Adabas.