

Test Programs



A test program is a Java client generated batch VADELIA program, the purpose of which is to batch test the modules of your application.

It can call up subroutines in your application (VADELIA batch or SADELIA) or consume Web services and check the results provided by means of the instruction [ASSERT](#).

In the event that the programs tested are not Java client generated objects, you will need to place their call in a server part corresponding to the platform where they are generated.

A test program contains four particular public client procedures (BEFORE_PGM, AFTER_PGM, BEFORE and AFTER) and one or more public client procedures performing the tests (hereinafter called "test procedures").

All these procedures have no parameters.

A test procedure contains one or more [ASSERT](#) instructions.

When running a test program, the system first calls the BEFORE_PGM procedure and then, for each test procedure, it calls the following sequence

call the BEFORE procedure, call the test procedure, call the AFTER procedure and then once all the test procedures have been called, call the AFTER_PGM procedure.

Each of the BEFORE_PGM, AFTER_PGM, BEFORE or AFTER procedures is optional.

You do not have control over the order in which test procedures are called.

Similarly, when executing a set of test programs, you do not control the order in which the test programs are executed.

The BEFORE_PGM and AFTER_PGM procedures share global variables and the same possible Middleware connection.

The procedures in the sequence "BEFORE procedure, test procedure, AFTER procedure" share global variables and the same possible Middleware connection, but this is not the case for two different "test" sequences or a test sequence and the BEFORE_PGM/AFTER_PGM procedures.

Each test program must be assigned to an Adelia [task](#) of the "test" type.

All your test programs can be distributed over one or more test-type tasks.

The test tasks are used to indicate a package and target generation directory for the Java client source specific to the programs of the task and thereby deviate from the values shown on the "Java Cclient" page for the attributes of the environment (public programs) or of the logic level of the application area (private programs).

Example of a test program:

In the example below, the test program tests the SADELIA PERSON-READ and _DEL programs of an application using PERSON-READ_EXISTS, PERSON-READ_DOES_NOT_EXIST and PERSON_DEL_OK test procedures.

The CONFIG_TEST management rule establishes a dynamic Middleware connection (via VaToolBx functions) by retrieving the connection information in the context of the test project.

The BEFORE_PGM procedure creates a test set which will be used by the test procedures.

The BEFORE procedure establishes just the Middleware connection (via the CONFIG_TEST management rule).

```

PGM
  DECL PGM
  INIT PGM
  BEFORE_PGM
    DECLARATION
  AFTER_PGM
    DECLARATION
  BEFORE
    DECLARATION
  PERSON_READ_EXIST
    DECLARATION
  PERSON_READ_NOT_EXIST
    DECLARATION
  PERSON_DEL_OK
    DECLARATION

```

```

PGM
  DECL PGM
    DSI DSPERSON *REF_LDM(PERSON)
    NUM_BIN_4 WRET_CODE
  INIT PGM
  BEFORE_PGM
    DECLARATION
    INSERT_MR CONFIG_TEST
    DELETE_SQL PERSON
    PID_NUMBER = 1
    PNAME = 'NAME001'
    PFIRSTNAME = 'FIRSTNAME0001'
    PBIRTH_DATE = &NUM_TO_DATE(20010214)
    CREATE_SQL PERSON
    PID_NUMBER = 2
    PNAME = 'NAME002'
    PFIRSTNAME = 'FIRSTNAME0002'
    PBIRTH_DATE = &NUM_TO_DATE(20020214)
    CREATE_SQL PERSON
    PID_NUMBER = 3
    PNAME = 'NAME003'
    PFIRSTNAME = 'FIRSTNAME0003'
    PBIRTH_DATE = &NUM_TO_DATE(20030214)
    CREATE_SQL PERSON
    COMMIT
  AFTER_PGM
    DECLARATION
    DELETE_SQL PERSON
    COMMIT
  BEFORE
    DECLARATION
    INSERT_MR CONFIG_TEST
  PERSON_READ_EXIST
    DECLARATION
    DATE WDATE
    PID_NUMBER = 1
    CALL PERSON_READ DSPERSON WRET_CODE
    ASSERT 0 WRET_CODE 'Return code 1'
    ASSERT 1 PID_NUMBER 'ID number 1'
    ASSERT 'NAME001' PNAME 'Name 1'
    ASSERT 'FIRSTNAME0001' PFIRSTNAME 'Firstname 1'
    WDATE = &NUM_TO_DATE(20010214)
    ASSERT WDATE PBIRTH_DATE 'Birth date 1'
    PID_NUMBER = 3
    CALL PERSON_READ DSPERSON WRET_CODE
    ASSERT 0 WRET_CODE 'Return code 3'
    ASSERT 3 PID_NUMBER 'ID number 3'
    ASSERT 'NAME003' PNAME 'Name 3'
    ASSERT 'FIRSTNAME0003' PFIRSTNAME 'Firstname 3'
    WDATE = &NUM_TO_DATE(20030214)
    ASSERT WDATE PBIRTH_DATE 'Birth date 3'
  PERSON_READ_NOT_EXIST
    DECLARATION
    PID_NUMBER = 5
    CALL PERSON_READ DSPERSON WRET_CODE
    ASSERT 100 WRET_CODE 'Return code'
  PERSON_DEL_OK
    DECLARATION
    DATE WDATE
    num_bin_4 WRET_CODE2
    PID_NUMBER = 2
    CALL PERSON_DEL PID_NUMBER WRET_CODE
    PID_NUMBER = 2
    CALL PERSON_READ DSPERSON WRET_CODE2
    ROLL_BACK
    ASSERT 0 WRET_CODE 'Return code delete'

```

```
|| | L L ASSERT 100 WRET_CODE2 'Return code read'
```

Source of CONFIG_TEST management rule:

MR

```
DECL ALPHA(50) adeliaPhysicalServerHost
DECL ALPHA(50) adeliaPhysicalServerPort
DECL ALPHA(50) adeliaPhysicalServerProfile
DECL ALPHA(50) adeliaPhysicalServerPassword
DECL ALPHA(50) adeliaLogicalServer
DECL ALPHA(50) adeliaDatabase
DECL BOOL ReturnCode

CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaPhysicalServerHost' adeliaPhysicalServerHost 50
CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaPhysicalServerPort' adeliaPhysicalServerPort 50
CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaPhysicalServerProfile' adeliaPhysicalServerProfile 50
CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaPhysicalServerPassword' adeliaPhysicalServerPassword 50
CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaLogicalServer' adeliaLogicalServer 50
CALL_CLASS 'VaToolBx' 'VaToolBxGetEnv' 'adeliaDatabase' adeliaDatabase 50

CALL_CLASS 'VATOOLBX' 'VaToolBxRemoveAssociation' adeliaLogicalServer ' ' ReturnCode
CALL_CLASS 'VATOOLBX.DLL' 'VaToolBxRemoveLogicalServer' adeliaLogicalServer ReturnCode
CALL_CLASS 'VATOOLBX.DLL' 'VaToolBxRemovePhysicalServer' 'ADELIA_TEST_SERVER' ReturnCode

CALL_CLASS 'VATOOLBX.DLL' 'VaToolBxAddLogicalServer' adeliaLogicalServer 'test' *TRUE ReturnCode
CALL_CLASS 'VATOOLBX.DLL' 'VaToolBxAddPhysicalServer' 'ADELIA_TEST_SERVER' 'test' adeliaPhysicalServerHost -
    adeliaPhysicalServerPort adeliaPhysicalServerProfile adeliaPhysicalServerPassword ReturnCode

CALL_CLASS 'VATOOLBX' 'VaToolBxAddAssociation' adeliaLogicalServer 'ADELIA_TEST_SERVER' adeliaDatabase ReturnCode
```

Note: we provide the source of the CONFIG_TEST management rule in the BDCADEL environment. This environment is in the full package of the Adelia Studio product.

[Top of page](#)