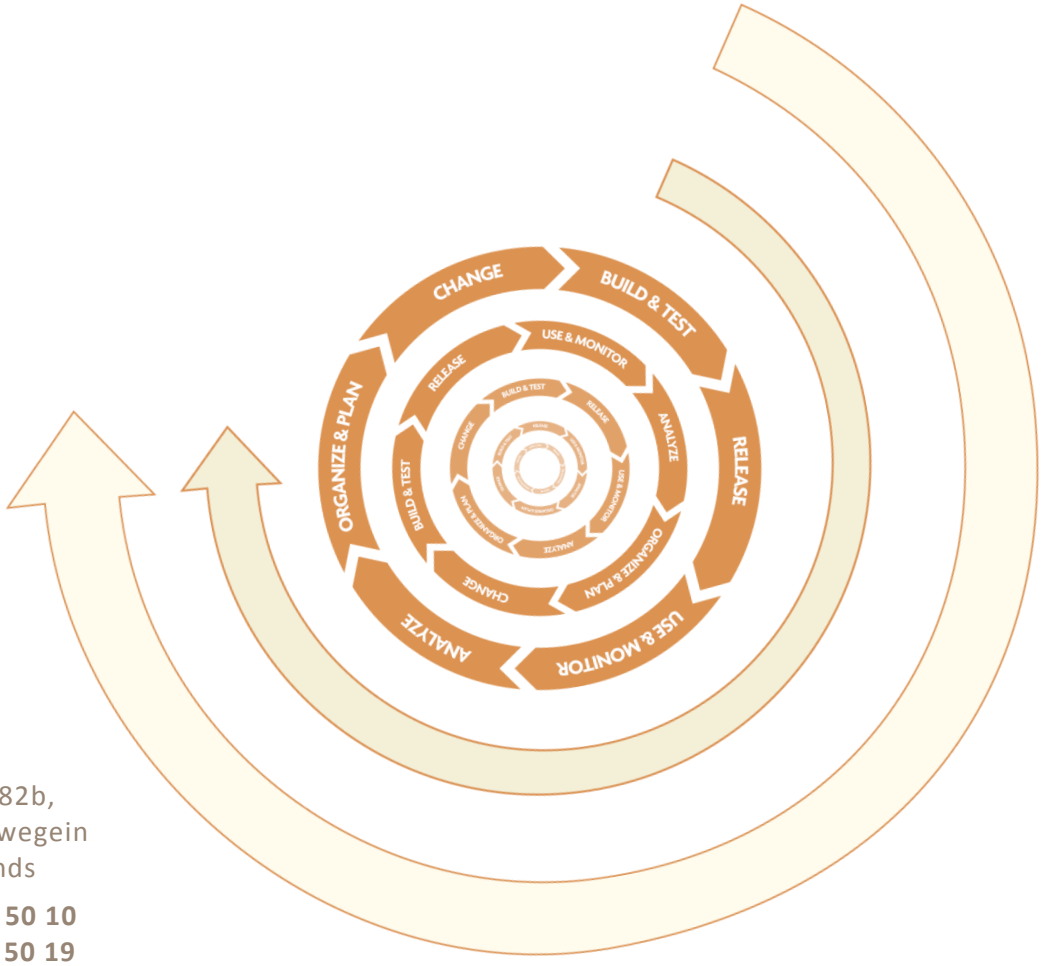
REMAIN SOFTWARE

TD/OMS

SOFTWARE CHANGE MANAGEMENT SOLUTION

Application Manager Training Casebook

Version 7



Remain B.V.
Dukatenburg 82b,
3437 AE Nieuwegein
The Netherlands

T + 31 30 600 50 10
F + 31 30 600 50 19
info@remainsoftware.com
www.remainsoftware.com

TABLE OF CONTENTS

CASE DESCRIPTION

BACKGROUND

One of Remains customers is using TD/OMS for quite some time now. With the help of TD/OMS, business applications are being managed and maintained. The IT-department of this organization has four developers (users OMSOP0***X***), one TD/OMS manager (user OMSOP0***X***), and a Request Manager (OMSOP05). Recently a new application has been developed. This application has to be set up in TD/OMS.
In all user profile names, the ***X*** should be replaced by the number of your group.

SPECIFICATIONS

The specifications of the concerning applications are the following:

Application name	OMS0***X***
Request Processor	OMSOP0**5**
Dump Library	OMSOP***X***DMP
Fall Back Save Library	OMSOP***X***F
Environments	*DEV (development), *TST (test), *ACC (acceptance) *PRD (production)
Libraries development	OMSOP***X*** D (used for all object types) OMSOP***X***DD (used for data (*FILE (PF/LF) and *DTAARA)) OMSOP***X***DS (used for sources)
Libraries test	OMSOP***X***T (used for all object types) OMSOP***X***TD (used for data (*FILE (PF/LF) and *DTAARA)) OMSOP***X***TS (used for sources)
Libraries acceptance	None (status environment)
Libraries production	OMSOP***X***P (used for all object types) OMSOP***X***PD (used for data (*FILE (PF/LF) and *DTAARA)) OMSOP***X***PS (used for sources) OMSOP (used for object relations only)

In all library names, the ***X*** should be replaced by the number of your group.

CHANGE MANAGEMENT RULES

The most important Change Management rules of the company:
Before an object is transferred to the production environment, the Request Manager has to give his approval with the use of ratification.
Whenever an object is being transferred from the production environment to the development environment, the Request Manager has to receive a message about this action

PURPOSE OF THIS CASE

By making the following exercises, you will learn how to add a new application to your TD/OMS system and how to define this application with the use of the most common functionalities in TD/OMS.

NUMBERING

The first part of the numbers of the exercises refer to the chapter in the course book. E.g. exercise 4.1 is related to chapter 4 in the course book.

EXERCISE 2

1. CREATING THE APPLICATION

In this exercise, you will add a new application.

- Start the function Application Definition.
- Press F6 to add a new application to the TD/OMS system.
- Fill in the name of the application.
- Fill in the person who is responsible for the request handling (the Request Manager).
- Fill in the initial request number. The request codes should start with the characters REQ.
- Fill in the initial fix number. The fix codes should start with the characters FIX.
- Make sure that the attributes and authorizations will be copied from the object that is being replaced in the target environment.
- Fill in the name of the dump library (This library will be used in other exercises).
- Fill in the name of the fall back save library (This library will be used in other exercises).

EXERCISE 3

1. ASSIGNING USERS

In this exercise, you will assign users to your application.

- Start the function User Authority Maintenance.
- Add all the users mentioned in the case description (refer to page 2 for details). The Request Manager and your own profile should be defined with authorization code 3, the developers with code 2, and other users with code 1.

2. ENVIRONMENTS AND LIBRARY LISTS

In this exercise, you will define the environments within your application.

- Start the function Environment Maintenance.
- Create the four environments (*DEV, *TST, *ACC en*PRD) as described in the case description. Define the *ACC (acceptance) environment as a *status* environment.
- Set the element ***Include Library Lists*** to *ALL and set up ratification for all transfers of requests from the acceptance environment to the production environment.
- Create a library list for each environment.
- Set the element ***Quit and fall-back*** for the production environment to on 1st failure.
- Add the correct libraries to each the library list. Refer to the case description (paragraph 0 on page 2) for details. Define *KEEP for the data library.

3. TD/OMS DATABASE

In this exercise, you will fill the TD/OMS object file, and examine the reports that are created in this process.

- Start the function *Fill Object File*.
- Fill the TD/OMS database with all objects from all environments.
- When the job has finished, examine the spool files that were created.

4. CREATING TABLES

In this exercise, you will define codes in the tables that are used within TD/OMS.

- Start the function *Table Maintenance*.
- Create the priority codes *HIGH, *NORM and *CALM. The code *NORM has to be the default.
- Create the reason codes *NOTF (Not found) and *CANC (Cancelled).
- Create the request types *PROJ (Project), *CLRQ (Client request), *INRQ (Internal request).
- Create the fix types *UNKN (Unknown), *FUNC (Functionality), *REPA (Repair).

5. RATIFICATION

In this exercise, you will define a ratification group, and you will add a user to this group.

- Start the function Ratification Group Maintenance.
- Create a ratification group called QC (Quality Control). Only the current application should be allowed to use the ratification group.
- Add the Request Manager to the ratification group QC. (Refer to page 2 for details).
- From the Main menu, choose option 2, option 4.
- Assign with option 13 the group QC to the acceptance environment.

EXERCISE 4

1. WORKING WITH OBJECTS

In this exercise, you will examine the information stored in the TD/OMS database.

- Start the function *Object Maintenance*.
- Check the list of objects that are registered for this application.
- Which objects does object CLPGM001A have a relation with?
- Press function key F6 to add a relation with object CLPGM012.
- What do you notice about the relation type?
- What is a relation type *I/O?
- Check the relations with the help of User Option DS (Display Source).
- What does the value *none in the column *Env* (environment) mean?

2. CREATING A REQUEST

In this exercise, you will create some new requests.

- Start the function *Request Maintenance*.
- Create two requests. Fill in the request type and the priority.
- On the screen *Work with Requests*, press function key F17 and examine the subset options.

3. CREATING A FIX

In this exercise, you will create some fixes.

- Start the function *Fix Maintenance*.
- Create a new fix. This fix will not be connected to a request, because the function *Fix Maintenance* was not started from within the function *Request Maintenance*. Fill in the fix type and the priority.
- From the Main menu, choose option 3, option 2. This will start the function *Request Maintenance*.
- Select option 12 in front of one of the requests. This will start the function *Fix Maintenance*.
- Create a new fix. This time, the fix will automatically be connected to the request. Fill in once again the fix type and the priority.
- From the Main menu, choose option 3, option 3.
- Connect the ‘stand alone’ fix to the other request with option 3 (Connect Request).
- Press function key F10 to examine the connections between requests and fixes.

4. CONNECTING OBJECTS

In this exercise, you will connect some objects to a fix.

- Start the function *Fix Maintenance*.
- Select option 12 in front of one of the fixes.
- Press function key F6. This will display the screen *Subset Object List*. Press Enter.
- Select two objects from the object list.
- Connect the selected objects to the fix.

5. TRANSFERRING OBJECTS

In this exercise, you will execute an object transfer.

- Start the function *Fix Maintenance*.
- Transfer the selected objects through the TD/OMS cycle.
- Start the function *Object Maintenance*.
- Check the history of the objects by selecting option 7 on the screen *Work with Objects*.
- Examine the transfer history by selecting option 20 on the screen *Work with Objects*.

6. CREATING AN EMERGENCY ENVIRONMENT

In this exercise, you will add an emergency environment to the application cycle.

- Start the function Environment Maintenance.
- Complete the application cycle with an Emergency environment.
- Connect the library OMSOPXEMG (X is the number of your group) to the environment. Make sure that the library can receive all kind of objects and sources.
- Set up security for the emergency environment (option 8) so that your group is allowed to work with emergency situations.

7. LOCK AND FOLLOW/LEADER

In this exercise, you will experiment with solution conflicts.

- Create a request.
- Connect the request to a fix.
- Connect an object of the production environment to the fix.
- Copy the solution to the development environment.
- Transfer the solution to the test environment.
- Create a new fix.
- Connect the same object from the production environment to the second fix (Select the option to connect this object and Lock the solution that is in the test environment).
- Transfer the second solution to the emergency environment.
- Check the status of the solution in the test environment.
- Transfer the solution from the emergency environment back into production.
- Transfer the solution from the test environment back into production.
- Do you notice anything?
- Try to solve the situation by “confirming” the transfer.

This (optional) exercise covers both Lock and Follow/Leader.

- Create a request.
- Connect the request to a fix.
- Connect an object to the fix.
- Copy the solution to the development environment.
- Transfer the solution to the acceptance environment.
- Create a new fix.
- Connect the same object from the production environment to the fix. Select the “Connect alternative and lock other solutions” option.
- Check the status of both solutions. What do you notice ?
- Move the solution of the second fix to the acceptance environment. Please check the help text of any additional screen shown before you continue.
- Check the status of both solutions. What do you notice ?
- Transfer the solution from the acceptance environment back into production.

EXERCISE 5

1. DUMPED SOURCES

In this exercise, you will work with dumped sources.

- Start the function Dumped Sources Maintenance.
- Examine the entries on the screen.
- Press function key F14 and find out what options are available for deleting dumped sources.

EXERCISE 6

1. CREATING A REMOTE LOCATION

In this exercise, you will create a remote location.

- Start the function Remote Location Maintenance.
- Press function key F6 to add a remote location. The name of the machine and the version of the operating system will be supplied by your trainer. Use for all job descriptions DMSJOBD (library OMSOP).
- Make sure that the application definitions and the application exceptions are also distributed.

2. DEFINE DISTRIBUTION

In this exercise, you will define distribution to the remote location that you have created in the previous exercise.

- Start the function Environment Maintenance.
- Create a remote library list in the *PRD environment.
- Set up distribution for the transfer from *ACC to *PRD.

EXERCISE 7

1. LOGGING OF TRANSFERS

In this exercise, you will examine the logs that have been generated by the transfers that were executed in the previous exercises.

- Start the function Display Application Log.
- Select an entry on the screen Work with Jobs. Examine the contents of the log.
- Press F14 to collapse the log.
- Select option 8 to display a collapsed section of the log.
- Start the function Fix Maintenance.
- Select option 20 to display the screen Work with Transfer History.

EXERCISE 8

1. EXCEPTIONS

In this exercise, you will create and test an exception.

- Start the function *Function Maintenance* (located on the menu *Exception Management*).
- Create an exception function. The function is a SNDMSG command to inform the Request Manager (see case description).
- Add an exception selection to the created function. The function should be executed at UEP 9, for object transfers from *PRD to *DEV.
- Create a fix and test the definitions you have made.

