



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

Training

ANSAwise - Simple Bank Exercise Guide

Chris Mayers

Abstract

Organizations wish to obtain practical experience in designing distributed applications. Within a training course, a hands-on exercise is the most effective way to provide that experience.

CNET have requested such an exercise for inclusion within their training programme for distributed systems.

This document describes the Simple Bank Exercise, which uses ANSAware 4.1. This document is intended for the course presenter, and describes the aims and objectives of the exercise, how the exercise is structured, and the points that are likely to arise from it.

(The components of this exercise may also be used in different ways in different training courses.)

APM.1547.01

Approved
Briefing Note

18th August 1995

Distribution:
Supersedes:
Superseded by:

ANSAwise - Simple Bank Exercise Guide



ANSAwise - Simple Bank Exercise Guide

Chris Mayers

APM.1547.01

18th August 1995

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by Architecture Projects Management Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

Architecture Projects Management Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK
INTERNATIONAL
FAX
E-MAIL

(01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk

Copyright © 1995 Architecture Projects Management Limited
The copyright is held on behalf of the sponsors for the time being of the ANSA Workprogramme.

Architecture Projects Management Limited takes no responsibility for the consequences of errors or omissions in this Report, nor for any damages resulting from the application of the ideas expressed herein.

1 Introduction

1.1 Audience

This document is intended for course presenters of the Simple Bank Exercise.

1.2 Scope

This version of the document is specifically intended for the CNET project [Proposal]. This version specifies the structure and content of the Simple Bank Exercise, but does not include detailed instructions for the course presenter (see §4 *Course presenter information*). This information will be added to this document as the exercise is developed.

1.3 Status

This is a draft for comment.

2 Overview

2.1 About the Simple Bank Exercise

This exercise is based on the standard Simple Bank sample application supplied with ANSAware [AWAP4.1]. It is developed from a similar exercise used in the ANSAware-specific [Course].

2.2 Aims of the Simple Bank Exercise

The aims of the Simple Bank Exercise are to demonstrate:

- that developing distributed applications is straightforward
- what a specification can and must contain
- the challenges of non-transparency for distributed applications

This exercise does not aim to teach the use of ANSAware itself; ANSAware is a means to an end. ANSAware 4.1.1 is used in this exercise because it provides all the features necessary to demonstrate distributed systems concepts, and because it is simple to learn.

2.3 Objectives of the Simple Bank Exercise

On completing the Simple Bank Exercise, participants will be able to:

- Specify an interface in ANSAware IDL
- Build simple ANSAware applications (client and server)
- Implement a client application using ANSAware PREPC
- Implement the server
- Add error-handling to a client application
- Add concurrency control to a server
- Use the ANSAware Trader and Factory services

This exercise is not intended for self-study by participants.

The following topics are not within the scope of this exercise:

- installing and configuring ANSAware
- porting ANSAware applications (between platforms)
- advanced testing and debugging
- replication
- use of the ANSAware real-time features (in ANSAware/RT)
- security
- persistent storage

2.4 Participant prerequisites

Participants in the Simple Bank Exercise must:

- have attended the course Introduction to Design of Distributed Systems
- have basic UNIX user skills (basic commands and use of a Unix text editor)
- be fluent in C
- have 2-3 years industrial software engineering experience

No previous experience with ANSAware is required. In fact, because this is a basic exercise, participants who have already had previous experience with ANSAware may not need to attend this exercise.

3 Structure of the Simple Bank Exercise

3.1 Overall structure

The Simple Bank Exercise takes 1 day of class time. It is intended to be delivered immediately following an Introduction to Design of Distributed Applications course; therefore only a minimal review of basic principles is required.

The exercise starts with a standard 10-minute Welcome session, and finishes with a standard 10-minute Roundup. The exercise proper is structured as five modules each lasting 1.5 hours. These modules are intended to be presented together, in the order specified below; they are not stand-alone, and cannot be used independently.

Normally two modules would be delivered in the morning, one module would span the lunch breaks, and two further modules would be delivered in the afternoon.

Each module consists of a standard training presentation (approximately 30 slides). Each presentation is in two halves; an introduction explaining what the participants will do, and a summary (debrief) explaining what the participants have learned. Each half of the presentation will last about 10 minutes, including discussion and questions/answers. Participants therefore have more than 1 hour to complete the activity.

Each module consists of a single activity. Each module starts with a new set of source code. Thus, whether participants succeed or fail to complete one activity, they all start on an equal basis on the next activity.

Participants work in pairs.

The modules are:

- Welcome
- Building Applications with ANSAware Tools
- Trading for Distributed Systems
- Exploiting Concurrency in Distributed Applications
- Implementing Robust Distributed Applications
- Managing Distributed Applications
- Roundup

3.2 Welcome

This brief session welcomes the participants and explains the purpose and structure of the course.

3.3 Building Applications with ANSAware Tools

This session:

- explains why ANSAware has been selected for this exercise
- gives some examples of systems which use ANSAware
- explains briefly how ANSAware compares with CORBA
- describes ANSAware IDL
- shows the steps to build an ANSAware client and server
- shows the overall structure of a simple ANSAware client and server
- shows how to run an ANSAware client and server

This session then explains the overall setting for the Simple Bank exercise, and why it has been chosen.

The participants are supplied with incomplete source code for Simple Bank. The Access operation of the Account interface does not contain the logic for validating the PIN against the account number. An outline is provided, with comments indicating how the operation should be implemented.

Participants then:

- complete the specification of the Banking/SBank interface in ANSAware IDL
- complete the implementation of a client of the Simple Bank service, in ANSAware PREPC, and C
- complete the implementation of the server in ANSAware PREPC, and C
- test the functioning of client and server

(Note that although the source code is incomplete, participants are supplied with complete makefiles, so building the client and server applications is straightforward.)

Note that the ANSAware Trader has not yet been described. Therefore the existence of the Trader must be transparent to the participants. Furthermore, the clients and servers are being modified by different participants; each participant's client must use their own server. This will require some specific configuration for each participant. (The following module will explain the functioning of the Trader.)

3.4 Using the ANSAware Trader service

In this session:

- the role of trading in ODP is reviewed
- the importance of type compatibility is described
- the ANSAware Trading service is described

Participants use the ANSAware Trader tools (both GUI and command-line) to:

- inspect service offers
- configure interface types in the Trader
- configure Trader contexts
- delete stale service offers (to recover from errors)

In this session, participants also:

- design the use of Trader properties by the Simple Bank service
- modify the Simple Bank client and server to use these properties

3.5 Exploiting Concurrency in Distributed Applications

This session explains:

- how concurrency can improve application performance
- transactions and the ACID properties
- simple concurrency control using locks
- implications of using concurrency features in C
- how to use ANSAware basic concurrency features (mutex locks)

Participants then:

- modify the Simple Bank server application to support multiple concurrent clients
- configure the Simple Bank server application to support multiple concurrent clients
- test the concurrent behaviour of the modified Simple Bank server

This session does not cover:

- the explicit use of ANSAware multi-threading (fork/join)
- asynchronous RPC (initiate/redeem)
- optimistic concurrency control techniques

Note that the ANSAware concurrency model is more sophisticated than that used by POSIX/DCE Threads (pthreads), and the traditional approach taken in database concurrency. However the basic concurrency features are the same, and this session concentrates on them.

3.5.1 Implementing Robust Distributed Applications

This session explains:

- why distributed applications must be able to handle a wider range of error conditions than centralized applications
- how to use ANSAware exceptions to handle these error conditions

Participants then modify the client application to:

- use ANSAware exception handling to report application-specific error messages for errors detected by the infrastructure (for example, when a server is not running)
- override the default relocation policy (for retries)
- report application-level error messages (for example, security violations)

Participants also test the behaviour of the modified client application when these error conditions occur.

The ANSAware exception handling mechanisms are not as powerful as ANSA terminations (or, indeed, CORBA exceptions). Furthermore, the language mappings for exceptions are very different (ANSAware does not use C

exception macros). This session concentrates on the principles of exception handling.

Note that this session does not cover memory management issues. Although this is an important aspect of implementing robust distributed applications, it is not within the scope of this exercise.

3.6 Managing Distributed Applications

This session explains

- the different methods of creating and activating a server
- the role of the Factory service
- the role of the Node Manager service, and its integration with the Trader
- managing the services with ANSAware

Participants then modify the Simple Bank server to:

- be manageable via the Factory and Node Manager
- handle new error conditions accordingly

Note that the Simple Bank source code shipped with ANSAware 4.1.1 does not support being managed; it is in fact suggested as an extension.

3.7 Roundup

This is a 10-minute session. It:

- summarizes what has been learned during the exercise
- lists some additional features of ANSAware that have not been covered by the exercise
- explains the evolution of ANSAware, in ANSAware/RT, and also towards CORBA
- explains how to find out more about ANSAware, and distributed systems products in general

Some of these topics will be covered in more detail in the Advanced Design of Distributed Systems course.

4 Course presenter information

4.1 Bill of materials

To be supplied.

4.2 Equipment requirements

To be supplied.

4.3 Presenter prerequisites

The person presenting this exercise must:

- have presented the Introduction to Design of Distributed Systems course
- be experienced with building ANSAware applications, and know how to recover from common errors
- be familiar with known problems with ANSAware 4.1.1
- complete the Simple Bank exercise themselves

Other presenter prerequisites are to be supplied.

References

[AWAP4.1]

ANSAware 4.1 - Application Programming in ANSAware
RM.102.02

[AWSM4.1]

ANSAware 4.1 - System Manager's Guide
RM.100.02

[AWSP4.1]

ANSAware 4.1 - System Programming in ANSAware
RM.101.02

[Proposal]

Proposal for Training in Distributed Systems [CNET]
APM.1450

[Course]

Writing Distributed Applications using ANSA and ANSAware
APM.1261

[Exercises]

ANSAwise - case studies and exercises
APM.1401

