

# **Oracle Tuxedo Application Rehosting Workbench**

Process Guide

11g Release 1 (11.1.1.3.0)

December 2011

**ORACLE®**

Copyright © 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

## Introduction

What is Rehosting? . . . . .	1-1
Product Overview . . . . .	1-1
Rehosting Projects . . . . .	1-1
Projects, Phases and Steps . . . . .	1-2
Project Environment . . . . .	1-2
Project Phase Overview . . . . .	1-3
Overview of Using the Product in a Project . . . . .	1-5

## Processes

Process List . . . . .	2-10
Project Definition . . . . .	2-10
Asset Generation . . . . .	2-10
Asset Cataloging . . . . .	2-11
Asset Rationalization (Conditional Step) . . . . .	2-12
Target External Specifications . . . . .	2-12
Architecture Development . . . . .	2-13
Data Conversion . . . . .	2-13
Pilot Conversion . . . . .	2-15
Balance Conversion . . . . .	2-16
Operations Environment Migration . . . . .	2-16
System Integration . . . . .	2-17
Maintenance Conversion . . . . .	2-17
Production Switch-over . . . . .	2-18

## The Simple App Application

Introducing the Simple App Application . . . . .	3-19
Purpose . . . . .	3-19

Composition. . . . .	3-19
Usage. . . . .	3-20
Structure. . . . .	3-20
Description of the Simple App Components . . . . .	3-20
List of Components by Type . . . . .	3-21
CICS Screens . . . . .	3-21
CICS Programs. . . . .	3-21
Batch Programs . . . . .	3-22
Transaction Codes . . . . .	3-22
Jobs. . . . .	3-22
Map Descriptions. . . . .	3-24
Program Descriptions . . . . .	3-28
CICS Program Descriptions. . . . .	3-28
Batch Program Descriptions. . . . .	3-29
Simple App Functionalities . . . . .	3-31
Batch Processing . . . . .	3-31
VSAM Customers File Initial Load . . . . .	3-31
VSAM Customer File Update . . . . .	3-32
Data Description. . . . .	3-33
Report Layouts. . . . .	3-35
Using the Simple App Application. . . . .	3-36
Viewing A Customer Record. . . . .	3-37
Updating A Customer Record . . . . .	3-38
Printing Customer Reports . . . . .	3-40
Adding New Customers. . . . .	3-40
Simple App APIs . . . . .	3-41
MAP APIs . . . . .	3-41
SEND . . . . .	3-41

RECEIVE .....	3-41
NAVIGATION APIs .....	3-42
RETURN .....	3-42
XCTL .....	3-42
ABEND .....	3-42
VSAM APIs .....	3-42
STARTBR .....	3-42
READ .....	3-43
WRITE .....	3-44
REWRITE .....	3-44
READNEXT .....	3-44
READPREV .....	3-44
ENDBR .....	3-45
DELETE .....	3-45
MISCELLEANOUS APIs .....	3-45
TIME .....	3-45
HANDLE CONDITIONS .....	3-45
Simple App Documentation References .....	3-46
Detailed Description of the CICS APIs .....	3-46



# Introduction

This chapter covers the following topics:

[What is Rehosting?](#)

[Rehosting Projects](#)

## What is Rehosting?

Rehosting is a way to preserve the expensive investments in business logic and business data trapped in proprietary hardware and software, while opening paths to future modernization by moving to an open and more extensible architecture.

## Product Overview

Refine for Z/OS Replatforming package provides automated migration tools to enable customers to replatform COBOL, JCL, DB2, VSAM files and related assets from an IBM DB2 mainframe environment to a UNIX environment with a Tuxedo transaction processor and an Oracle database.

## Rehosting Projects

Refine for Z/OS Replatforming and Oracle Tuxedo Application Runtime for CICS and Batch are used within the context of a rehosting project. The process guide gives a global view of rehosting and the use of the conversion and runtime tools in this process. A rehosting project requires the

creation of specific test, integration and production environments. The different functions of a rehosting project are typically:

- Plan the project, setting the general requirements and planning the target architecture options.
- Prepare the assets to be converted and install and configure the target environment.
- Convert the assets.
- Integrate the assets in the target environment.

These functions may be performed iteratively, typically a project consists of the following phases:

- Setting project strategy.
- Assessment study.
- Pilot project.
- Implementation.

Each of these phases is made up of different steps, the results of these steps may be tested and the steps reiterated as necessary.

## Projects, Phases and Steps

Rehosting is performed within a project organized into phases and steps. Each step produces one or more deliverables. In parallel to the different steps of the rehosting project are a parallel series of test steps that validate the different phases of the project.

A project concerns different people who have different roles and responsibilities within the project. A project is carried out within an environment and it is impossible to describe the different phases and steps of a project without first describing the environment in which they need to be performed.

## Project Environment

There are five clearly distinct environments necessary to carry-out a rehosting project. These include two source (pre-migration) environments and three target (post-migration) environments as described below:

1. A current *production source* environment for the assets to be converted.



2. A *test source* environment for storing and isolating the operations extracted from the production environment and for creating a test database.
3. A *test target* environment for running, tuning and testing the converted assets.
4. An *integration target* environment, which is used to host all activities such as integration, operations migration or pre switch-over processing.
5. A *production target* environment for the converted and tested assets.

**Table 1-1 Project Environments**

Environments	Production	Integration	Test
Source	Production Source		Test Source
Target	Production Target	Integration target	Test Target

Depending on your needs, there may be several occurrences of the same platform to allow teams to work in parallel within the same project.

## Project Phase Overview

A project is divided into different phases of work, producing clear deliverables that are validated before proceeding to the next phase of the project. These phases are as follows

**Table 1-2 Project Phases**

Phase	Migration Steps	Test Step	Environment	Language/ Data
PROJECT MANAGEMENT	Project Definition			

**Table 1-2 Project Phases**

<b>Phase</b>	<b>Migration Steps</b>	<b>Test Step</b>	<b>Environment</b>	<b>Language/ Data</b>
PRE-CONVERSION	Asset Generation		Source Production to Source Test	Both
	Asset Cataloging Asset Rationalization (if necessary)		Source Test	Language
	External Specifications Architecture Development		Planning and preparation for Target environments	
	Data Conversion		Source Test to Target test	Data
TEST PREPARATION		Test Engineering Test Preparation Test Tooling	Source Test	Both
PILOT	Pilot Set Conversion	Pilot Set Testing	Target Test	Language
BALANCE	Balance Conversion	Balance Testing	Target Test	Language
INTEGRATION	System Integration Operations Migration	System Testing Operations Testing	Target Test to Target Integration	Both
SWITCH-OVER	Maintenance Conversion Maintenance Conversion Testing	Maintenance Testing Dry Runs	Target Test	Both
	Switch-Over		Target Integration to Target Production	
EDUCATION	Education And Training			

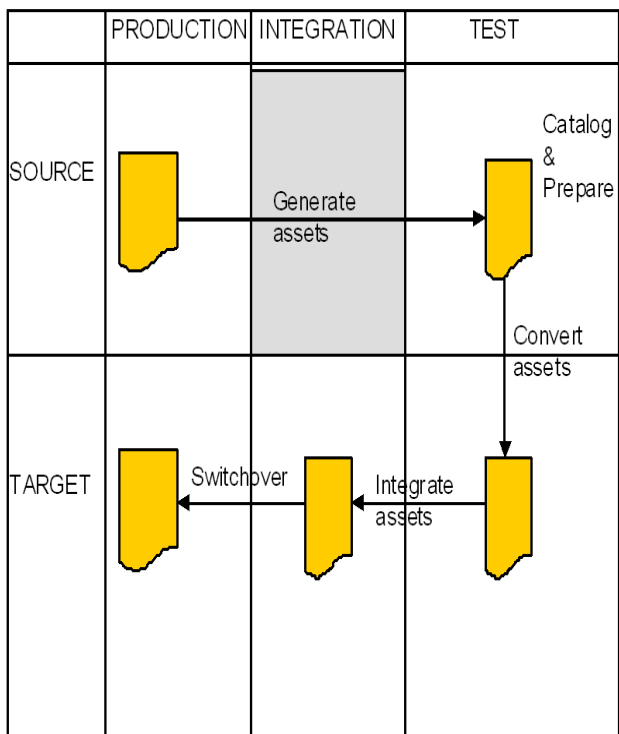
Graphically, the different phases may be grouped as shown below:

**Figure 1-1 Project Phases and Processes**

Phase	Process
Project Management	Project Definition
Pre-Conversion	<div>External Specifications</div> <div>Architecture Development</div> <div> <div>Data Discovery &amp; Download</div> <div>Asset Generation</div> </div> <div> <div>Data Remodeling &amp; Reload</div> <div>Asset Cataloging</div> </div> <div> <div>Data Validation</div> <div>Asset Rationalization</div> </div>
Pilot	Pilot Conversion
Balance	Balance Conversion
Integration	<div>System Integration</div> <div>Operations Integration</div>

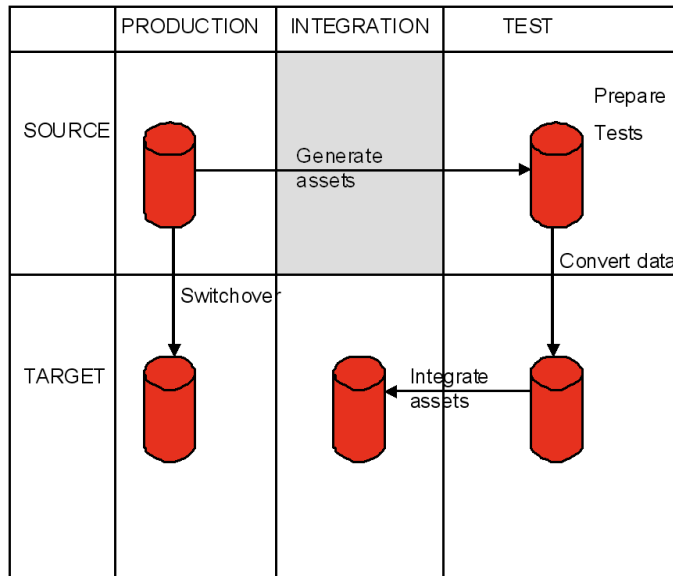
## Overview of Using the Product in a Project

Refine for Z/OS Replatforming is used for converting and integrating program components and data. The following diagram shows the use of the program components, and how they are used to prepare source files for migrating to different environments.

**Figure 1-2 Language Migration**

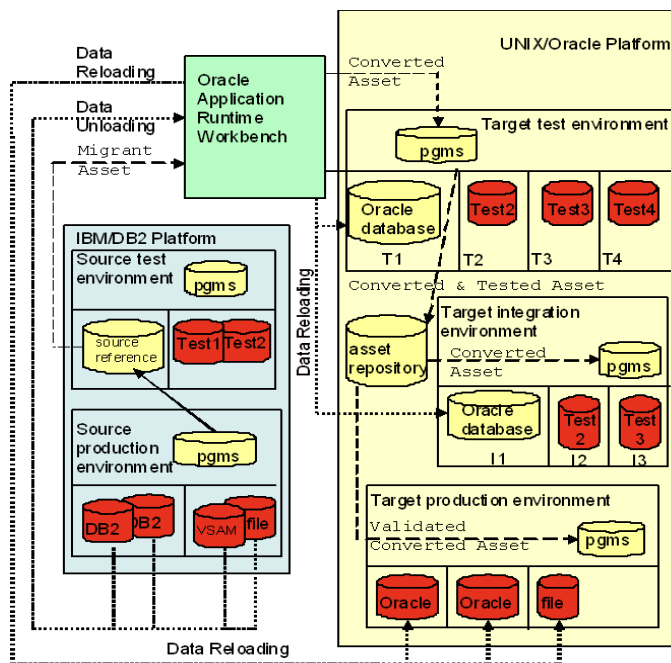
Refine for Z/OS Replatforming components are used to convert assets, enabling them after post-conversion adjustments to be moved from the source test environment to the target test environment.

Oracle Tuxedo Application Runtime for CICS and Batch components are used to integrate the converted assets after testing and integration preparation on the target test environment. The COBOL programs, JCL and associated components are integrated with the UNIX, Oracle database and Tuxedo transaction environments. These components are tested to work with the batch and CICS components produced during the conversion process.

**Figure 1-3 Data Migration**

The assets to be migrated are generated and moved to the source test machine. The migrated assets are then converted and moved to a target test machine containing an Oracle database. After testing, the data is moved to an integration environment where rehosting tools are integrated with the programs to test their quality and performance using the converted data. When switchover occurs the latest data is converted directly from the source production environment to the target production environment.

Figure 1-4 Migration Architecture



# Processes

This chapter describes the following processes:

- [Project Definition](#)
- [Asset Generation](#)
- [Asset Cataloging](#)
- [Asset Rationalization \(Conditional Step\)](#)
- [Target External Specifications](#)
- [Architecture Development](#)
- [Data Conversion](#)
- [Balance Conversion](#)
- [Operations Environment Migration](#)
- [System Integration](#)
- [Maintenance Conversion](#)
- [Production Switch-over](#)

# Process List

## Project Definition

### Purpose

Plan and prepare the work to be performed.

### Main Tasks

The main tasks are:

- Create the hardware and software environments required and prepare the human resources for the project
- Prepare a workplan including:
  - Detailed planning of the project.
  - Environment and technical requirements.
  - Assignment of resources.
  - Definition of milestones and critical path.
  - Risk mitigation plan.

## Asset Generation

This step moves all of the program components, production JCL and the data located in the *source production* environment to the *source test* environment.

### Purpose

- Identify and gather the asset information that will be migrated from the *source production* environment.
- Freeze and isolate the asset in a *source test* environment.
- Prepare this asset information for delivery to the production test environment.

### Main Tasks

The main tasks to be conducted are as follow:



- Identify the component version levels to be migrated.
- Create and initialize an occurrence of a *source test* environment.
- Copy the data and program components into the *source test* environment.
- Regenerate all asset executables, including compilations and linkage operations.
- Based on the analysis of delivered assets, select pilot assets: programs, copies, includes and tables to be included for compilation and linkage.
- Format the sources onto a suitable media for delivery.
- Execute and validate the source delivery process from the *source test* environment to the target environment.

## Asset Cataloging

### Purpose

- Define the precise set of components to be converted.
- Verify the completeness and consistency of the assets.

### Main Tasks

The main tasks to be conducted are as follow:

- Accept the delivered assets.
- Report on the extracted information.
- Construct an asset inventory including identification of dependencies outside of the asset set.
- Verify the completeness and accuracy of the provided source components.
- Improve the understanding and knowledge of the assets to be migrated, specifically regarding the interdependence between components (transactions, physical files...)
- Correction of the asset set.
- Validation of the cataloging results.
- Choice (or validation of choice) for the *pilot* set.

## Asset Rationalization (Conditional Step)

This process is only necessary when the asset cataloging has found too many errors.

### Purpose

- Constitute an asset set that meets the three quality measurement levels of: completeness, precision and consistency.

### Main Tasks

The main tasks to be conducted are as follow:

- Correction of the asset set (addition, modification, elimination).
- Cataloging of the new asset source.
- Measure and qualification of the three quality measurements criteria.

## Target External Specifications

### Purpose

Provide the following detailed external specifications:

- The target technical environment.
- The results of the conversion (schema, screens, language).

### Main Tasks

The principal external specification tasks are:

- Analyze the asset set and the source environment:
  - Pilot Code walk-through.
  - Source performance review.
- Study the asset set's technical specifications on the *source* and *target* environments.
- Define the external specifications of each *target* technical feature: hardware and software configuration, overall technical architecture and architecture implementation.
- Detailed definition of the asset set's external interfaces.

- Specifications of tools and adaptations to the conversion process
- Definition of the delivery conditions of the converted asset set.

## Architecture Development

### Purpose

- To set up and validate the *target* technical architecture, Oracle database, UNIX OS, Tuxedo and customer specific features.
- To deliver a prototype configuration that enables to evaluate both a critical batch and transaction process.
- Measure the response time in regard to performance expectations.

### Main Tasks

Provide the following detailed external specifications:

- Detail the specifications required for:
  - Developments:
    - Related to the source assets to convert.
    - Site specificities.
  - Adapting pre-existing technical components.
- Define *target* specifications of the technical components of the TP and batch architectures.
- Install the *target* standard architecture.
- Develop site-specific technical components based upon the requirements detailed above.
- Integrate and unit test technical components; batch and TP architecture.
- Integrate and aptitude test asset components.

## Data Conversion

Three steps compose the global data conversion process:

- The data discovery and downloading step.

- The data remodeling and reloading step.
- The data static and dynamic validation step.

## Purpose

### Data Discovery and Downloading

- Produce the source data unloading programs.
- Discover, define, clean and unload the source data.

### Data Remodeling and Reloading

- Define, create and validate the Oracle target environments (data base and files).
- Produce the data conversion tools.
- Transfer the *source* data system to the *target* system.

### Data Validation

- Validate static data to control the migrated data using process execution.
- Validate dynamic data to control the access control flow to the migrated data using process execution.

## Main Tasks

The main tasks include:

### Data Discovery and Downloading

- Identify the data environments of the assets to be migrated.
- Deliver COBOL data descriptions.
- Document the data assets.
- Generation of routines to qualify the data assets.
- Generation of routines to download the data assets including data transfer to target system.

### Data Remodeling and Reloading

- Study and specify the Oracle *target* data model.

- Use Refine for Z/OS Replatforming to generate reloading tools.
- Use Refine for Z/OS Replatforming to generate *target* data access routines.
- Reload *target* data.

## Data Validation

- Test conversion of drivers in the *target* environment.
- Execute programs within both *source* and *target* environment.
- Compare results and identify defects.
- Correct any defects found.

## Pilot Conversion

### Purpose

- Adapt and establish, with Refine for Z/OS Replatforming, the processes that will achieve during the project:
  - A very high accuracy in conversion.
  - A consistently good result.
- Implement and validate the *target* technical environment
- Produce the first conversion subset (when applicable).
- Validate the performance (load, delivery time and quality) of the implemented process and conversion tools.

### Main Tasks

The principal tasks to achieve are:

- Develop and fine tune the conversion rules that will produce the *target* components.
- Setup the *target* architecture software environment including Oracle database.
- Convert the whole asset set according to the industrial process.
- Extract and integrate the pilot onto the *target* environment.

- Perform an initial validation of the conversion rules through aptitude tests.
- Identify possible adjustments regarding project, environment, tools and processes.

## Balance Conversion

### Purpose

Implement the conversion process for the balance of assets remaining after the pilot conversion using the conversion rules and validated processing used for the pilot set with the same precision and iterative specifications.

### Main Tasks

- Conversion of asset balance using the industrial process validated with the pilot set.

## Operations Environment Migration

### Purpose

To redefine the production means and procedures for the *target* environment from the ones that are available in the *source* environment and in line with the requirements.

### Main Tasks

The relevant tasks are divided as follows:

- Analyze existing components: identify and analyze all components implemented to run in the *source* system and analyze relevant documentation.
- Define detailed specifications regarding each *target* operations component and identify constraints.
- Produce a detailed definition of each component of the operations sub-set and the choice of tools to use.
- Set up and configure an approved job scheduler to work with the converted JCL.
- Adapt the converted JCL to work with the approved job scheduler.
- Define a detailed execution schedule.
- Purchase or develop new operations components.

- KSH development, tools installation, ...
- Develop a production procedures guide in the *target* environment.
- Implement and validate each operations component.

## System Integration

### Purpose

- Define detailed integration specifications for each integration subset.
- Define an integration strategy and plan for each integration subset.
- Define a detailed executions plan.
- Develop integration tests.
- Perform integration tests.

### Main Tasks

- Define detailed integration specifications for each integration subset.
- Define integration strategy and plan for each integration subset.
- Define detailed executions plan.
- Test and measure the performance of the migrated applications in the *target* environment – optimize.
- Integrate execution tests.

## Maintenance Conversion

### Purpose

Convert all changes (addition, modifications, and deletions) which occurred between two images of the same asset (total or subset) with the following constraints:

- Produce a very high quality asset set to minimize the time and workload for validation.
- Minimize the conversion time to shorten the maintenance freeze period.

## Main Tasks

- Prepare maintenance conversion:
- Create a Maintenance Guide
- Plan and prepare maintenance changes
- Pre-deliver a maintenance image.
- Catalog.
- Update converted permanent data:
  - Data discovery, unloading and reloading.
- Update converted sources:
  - Source conversion.
- Execute aptitude tests.
- Execution, comparison and diagnosis.

## Production Switch-over

### Purpose

Uninterrupted switch over from the *source* to the *target* system, complying with the constraints (the size of the switch-over time window specifically).

### Main Tasks

The main tasks to achieve are:

- Define a Switch-over plan describing the global switch-over procedures.
- Develop a *source* to *target* data transfer procedure compatible with the delivery time constraints.
- Develop a contingency plan.
- Develop a detailed switch over plan.
- Validate the detailed plan by a switch-over dry run.
- Execute final switch over.



# The Simple App Application

## Introducing the Simple App Application

### Purpose

The Simple App application is intended to be used:

- To validate Refine for Z/OS Replatforming installation.
- To validate the whole migration process – from the source platform to the target platform.

### Composition

The simple Sample Application contains batch and online processing coded in the following languages:

- COBOL for batch and online programs.
- CICS for online programs.
- VSAM in batch and online program.
- JCL for jobs (plus the IDCAMS and SORT utilities).

Each program is intended to be as simple as possible, the purpose of this application is not to demonstrate all of the features of Refine for Z/OS Replatforming .

## Usage

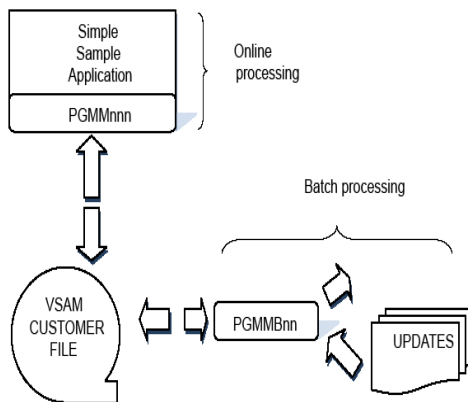
The overall process for using the Simple App application is:

1. Install Refine for Z/OS Replatforming environment.
2. Install the Simple App source platform files.
3. Convert the Simple App components.
4. Install Oracle Tuxedo Application Runtime for CICS and Batch environment.
5. Run the Simple App batch programs.
6. Run the Simple App transactional programs.

## Structure

The Simple App has the following structure:

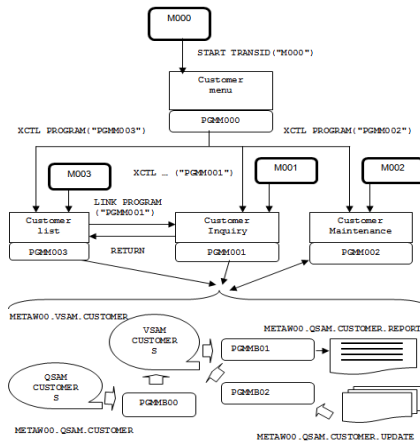
**Figure 3-1 Simple App Structure**



## Description of the Simple App Components

This section describes the Simple App component architecture and the online and batch interactions.

### Figure 3-2 Simple App Component Architecture



## List of Components by Type

## CICS Screens

Name	Description
MAPM000	Customer maintenance entry menu.
MAPM001	Customer data inquiry screen.
MAPM002	Customer data maintenance screen (new customer, update and delete customer).
MAPM003	Customer list screen.

## CICS Programs

Name	Description
PGMM000	Customer maintenance entry program.
PGMM001	Customer data inquiry program.

Name	Description
PGMM002	Customer data maintenance program (new customer, update and delete customer).
PGMM003	Customer list program.

## Batch Programs

Name	Description
PGMMB00	Initial load of the VSAM file. This file is loaded from a QSAM file.
PGMMB01	This program produces a list of the customers stored in the VSAM customer file.
PGMMB02	This program accesses a QSAM file containing commands and data used to update the main VSAM customers file.

## Transaction Codes

Name	Description
M000	Main entry transaction code (program PGMM000).
M001	Customer inquiry (program PGMM001).
M002	Customer maintenance (program PGMM002).
M003	Customer list (program PGMM003).

## Jobs

Name	Description
DEFVCUST	This job runs an IDCAMS utility in order to DELETE & DEFINE the VSAM customer file.
CHKVCUST	This job runs an IDCAMS utility and REPRO the VSAM file into a QSAM which can be easily read and checked.

Name	Description
LODVCUST	This job runs the PGMMB00 batch program which reads a sequential file containing data to be stored in the newly defined <i>VSAM</i> customer file.
PRTVCUST	This job runs the PGMMB01 batch program which produces a report of the customers referenced in the <i>VSAM</i> file.
UPDVCUST	This job runs the PGMMB02 batch program which contains data used to update the <i>VSAM</i> customer file.



**Listing 3-2 MAPM003 - Customer List**

```

.....1.....2.....3.....4.....5.....6.....7.....8
MAPM003          Customers list                      07/17/2009
M003

```

```

-----
Sel ID          Last name                      First name          Birth date
-----

```

```

@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@
@ @@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@@@@@@@@@@@@@@@@@@ @ @/@/@

```

```

-----
-PF7: Page up -PF8: page down -PF3: return to previous menu
-Info: Enter a non blank character and press enter to view customer's data
-----

```

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

### Listing 3-3 MAPM001 - Customer Detailed Information

---

```
.....1.....2.....3.....4.....5.....6.....7.....8
MAPMM01          Customer's detailed information          07/17/2009
M001
```

```
-----
Customer's identifier: àààààà
-----
```

```
-Last name____: àààààààààààààààààààààààààààààà
-First name____: àààààààààààààààààààà
-Address_____: àààààààààààààààààààààààààààààà
-City_____: àààààààààààààààààààà
-State_____: àà
-Birth date____: àà/àà/àààà
-Email_____: àààààààààààààààààààààààààààààààààà
-Phone number__: àààààààààà
```

```
-----
-PF03: return to previous screen
-----
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```



### Listing 3-4 MAPM002 - Customer Maintenance

```

.....1.....2.....3.....4.....5.....6.....7.....8
MAPM002           Customer's maintenance           07/17/2009
M002

```

Customer's identifier: àààààà                      Action: @@@@@@

[illegible]

```
-PF11: clear screen
-PF12: confirm action
-PF03: return to previous screen
```

[illegible]

## Program Descriptions

### CICS Program Descriptions

#### **PGMM000 (customer maintenance main menu)**

PGMM000 is the Simple Sample Application (access) main menu program. From this screen, one may access the following functions:

- Customer list
- Customer detailed information inquiry
- Customer data maintenance:
  - Add a new customer
  - Update customer information
  - Delete an existing customer

Each function can be reached by entering a specific function key. At the end of the called function, a return is performed back to this main menu.

Controls:

This program checks that a valid customer identification number is entered when accessing the inquiry and maintenance functions.

#### **PGMM001 (customer detailed information inquiry)**

This program receives – thru the communication area – a customer identification number. The program accesses the `VSAM` customer file in order to display the customer data when known.

Controls:

Assuming that no data can be modified by this program, no control is performed.

#### **PGMM002 (customer data maintenance)**

This program enables the maintenance of the `VSAM` customers file:

- Creating a new customer
- Updating an existing customer
- Deleting an existing customer

PGMM002 receives an action code and, when needed, a customer identification number. The action to be performed is displayed on the screen (CREATE, UPDATE or DELETE).

Controls:

The controls carried-out depend on the action to perform.

When creating a new customer, the programs checks that:

- The Customer identification number is not null, is numeric and not already used in the customer VSAM file.
- Last name, first name, address, city, email address and birth date are not equal to spaces.
- Phone number, if not equal to zero, must be numeric.

When updating an existing customer, the controls are the same as the ones used when creating a customer except for the customer identification number, which cannot be modified.

There is no control performed on a customer deletion request.

### **PGMMM03 (customer list)**

This program lists the customers stored in the VSAM file. A limited number of customers are displayed on each screen – the PF7 and PF8 keys give access to the previous and next pages. A non-blank character at the beginning of a customer's line gives access to the detailed customer information screen (see program PGMM001).

## **Batch Program Descriptions**

### **PGMMB00 (VSAM file initial load)**

This batch program reads a sequential file containing data to be stored in the VSAM file. For each record in the entry file, a VSAM record is created. The VSAM file must have been DELETED and DEFINED prior to executing this program. All data from the entry file is supposed to be valid: hence, no control is performed within this program.

### **PGMMB01 (customer report)**

PGMMB01 reads all the customer record from the VSAM file and produces a report. An example of the report produced is included in this document.

### **PGMMB02 (batch customer maintenance program)**

Each input sequential file record contains a three characters action code (ADD, UPD or DEL) and the customer data required when creating a new customer or updating an existing one. Each input line updates the *VSAM* customer file. Customer data from the sequential file is supposed to be valid so that no control is performed by the PGMMB02 batch program.

# Simple App Functionalities

## Batch Processing

Batch programs and utilities are used to populate, update and list the `VSAM` customer file that hosts the data managed within the application. Five jobs are delivered with the application. This chapter describes how they are used.

### VSAM Customers File Initial Load

1. Step 1: `VSAM` file definition

First, define the `VSAM` customer file thru the IDCAMS utility using `DELETE/DEFINE` commands.

2. Step 2: `VSAM` file initial load

The batch program `PGMMB00` is used to populate the `VSAM` file.

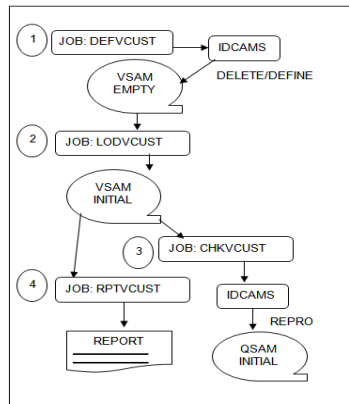
3. Step 3: `VSAM` file check

IDCAMS is used to `REPRO` the `VSAM` file into a sequential file which can easily be read in the `z/OS` environment.

4. Step 4: `VSAM` file report

Batch program `PGMMB01` produces a list of the customers stored in the `VSAM` customer file.

**Figure 3-3 VSAM Customers File Initial Load**



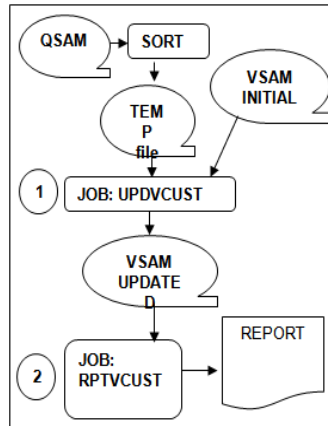
## VSAM Customer File Update

### 1. Step 1: VSAM file update

Batch program PGMMB02 reads a sequential file containing updates to be performed on VSAM customers file. The input file is sorted by the IBM standard sort utility.

### 2. Step 2: VSAM file report

Batch program PGMMB02 is used again in order to check that the updates were correctly applied to the VSAM customers file.

**Figure 3-4 VSAM File Update**

## Data Description

Simple Sample Application relies on only one VSAM main customer file. In order to keep the migration process as simple as possible, the data is stored in displayable formats (in COBOL syntax, this means that only "X" and "9" USAGE DISPLAY pictures are used).

The VSAM customer file contains the following information:

**Table 3-1 VSAM Customer File Description**

Name	Type	Length	Description
CUSTIDENT	Num.	6	Customer identification number
CUSTLNAME	Alpha	30	Customer last name
CUSTFNAME	Alpha	20	Customer first name
CUSTADDRS	Alpha	30	Customer address (street...)
CUSTCITY	Alpha	20	City
CUSTSTATE	Alpha	2	State
CUSTBDATE	Num.	8	Customer birth date

**Table 3-1 VSAM Customer File Description**

Name	Type	Length	Description
CUSTEMAIL	Alpha	40	Customer email address
CUSTPHONE	Num.	10	Customer phone number

This VSAM file is a Key Sequenced Data Set (KSDS) based on the customer's identification number. Each record contains 266 characters and the key is stored from position 1 to position 6.

### VSAM Customer File COBOL Description

**Listing 3-5 VSAM Customer File COBOL Description**


---

```

01  ODCSF0-RECORD.
    05  CUSTIDENT      PIC 9(006) .
    05  CUSTLNAME      PIC X(030) .
    05  CUSTFNAME      PIC X(020) .
    05  CUSTADDRS      PIC X(030) .
    05  CUSTCITY       PIC X(020) .
    05  CUSTSTATE      PIC X(002) .
    05  CUSTBDATE      PIC 9(008) .
    05  CUSTBDATE-G    REDEFINES CUSTBDATE.
    10  CUSTBDATE-CC   PIC 9(002) .
    10  CUSTBDATE-YY   PIC 9(002) .
    10  CUSTBDATE-MM   PIC 9(002) .
    10  CUSTBDATE-DD   PIC 9(002) .
    05  CUSTEMAIL      PIC X(040) .
    05  CUSTPHONE      PIC 9(010) .
    05  FILLER         PIC X(100) .

```

---



## Report Layouts

The following report is produced by program PGMMB01 that lists the customers from the VSAM file (META00.VSAM.CUSTOMER) after the initial load.

### Listing 3-6 Simple App Initial Report

---

```
PGMMB01                      Simple Sample Application                      07/16/2009
```

```

  _ ID _ _ LAST NAME      _ _ FIRST NAME  _ _ CITY      _ _ PHONE  _ _B. DATE _
-----
    1 Richardson          Bobby      New Orleans  5553557901 09/07/1961
    2 Roberts             Sammy      San Francisco 5559827383 01/24/1973
    3 Douglas             Burt       Atlanta      5556531100 10/12/1981
    4 Ewing               Samantha   New York     5558762763 07/27/1962
    5 Prince              Anne       Fresno       5553410156 12/25/1991

```

---

```
PGMMB01                      Simple Sample Application                      07/16/2009
```

```

  _ ID _ _ LAST NAME      _ _ FIRST NAME  _ _ CITY      _ _ PHONE  _ _B. DATE _
-----
    6 Colombus           Christopher Colombus  5557811021 07/27/1962
    7 Raul                Menedez    Fresno       5558981572 07/27/1962
    8 Doors               Bill       Seattle      5553122000 01/01/1958
    9 Awing               Charles     San antonio  5559990123 06/29/1929

```

---

The following report is based upon the updated customer file.

### Listing 3-7 Simple App Updated Customer File Report

---

```
PGMMB01                      Simple Sample Application                      07/16/2009
```

```

  _ ID _ _ LAST NAME      _ _ FIRST NAME  _ _ CITY      _ _ PHONE  _ _B. DATE _
-----

```

The Simple App Application

1	Richardson	Bobby	New Orleans	5553557901	09/07/1961
2	Roberts	Sammy Jr	San Francisco	5559827383	01/24/1973
3	Douglas	Burt	Atlanta	5556531100	10/12/1981
4	Ewing	Samantha	New York	5558762763	07/27/1962
5	Prince	Anne	Fresno	5553410156	12/25/1991
_ ID _	_ LAST NAME _	_ FIRST NAME _	_ CITY _	_ PHONE _	_ B. DATE _
-----					
6	Colombus	Christopher	Colombus	5557811021	07/27/1950
8	Doors	Bill	Seattle	5553122000	01/01/1958
9	Awing	Charles	San antonio	5559990123	06/29/1929
10	Simms	Arthur	New Orleans	5551298373	01/17/1969
11	LaFayette	Eric	Plesanton	5554653213	02/12/1995
12	Jackson	Mic	Fresno	5559800727	01/01/1959

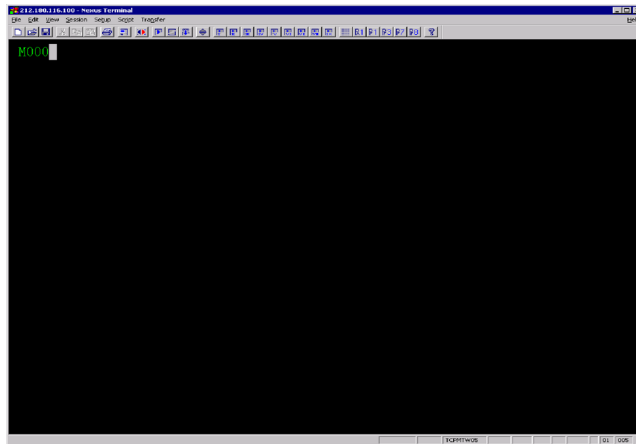
---

# Using the Simple App Application

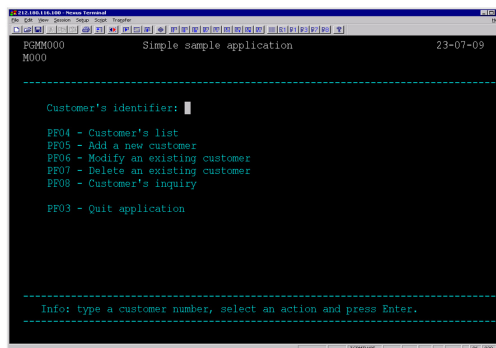
This section provides an example use of the Simple App Application in order to illustrate how to use the application.

## Viewing A Customer Record

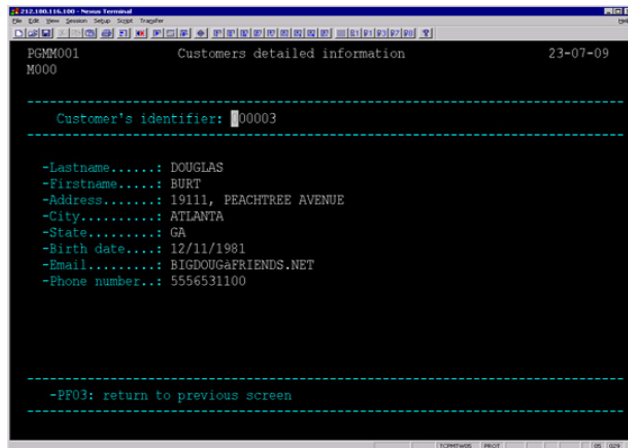
1. Access your CICS environment and enter the Simple App main transaction code M000 to connect to the application.



2. The Main menu is displayed. Enter 000003 in the Customer identifier field and press the PF08 key to inquire on the customer.

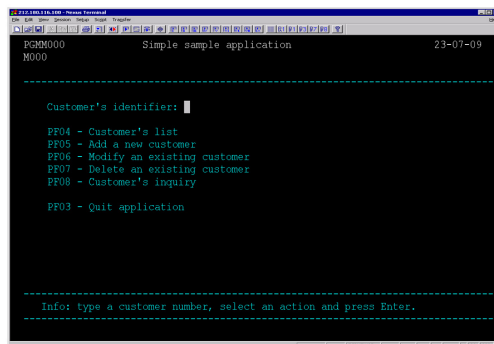


3. The customer record of Douglas Burt is displayed. Press PF03 to return to the Main menu.



## Updating A Customer Record

1. From the Main menu enter 000003 in the Customer identifier field and press the PF06 key to update customer information.



2. In the update screen all data may be modified. The screen header shows the action UPDATE. Change the customer's address and press ENTER.

```

212.100.116.100 - Nexus Terminal
PGMM002      Customers detailed information      23-07-09
M000

-----
Customer's identifier: 000003   Action: UPDATE
-----

-Lastname.....: DOUGLAS
-Firstname.....: BURT
-Address.....: 19111, PEACHTREE AVENUE
-City.....: ATLANTA
-State.....: GA
-Birth date....: 12/11/1981 (mm/dd/yyyy)
-Email.....: BIGDOUGAFRIENDS.NET
-Phone number..: 5556531100

-----
-PF03: return to previous screen
-----

```

3. Press PF12 to confirm the updates. A Maintenance OK message is displayed in the screen footer. To cancel your input, press PF03 in order to return to the Main menu.

```

212.100.116.100 - Nexus Terminal
PGMM002      Customers detailed information      23-07-09
M002

-----
Customer's identifier: 000003   Action: UPDATE
-----

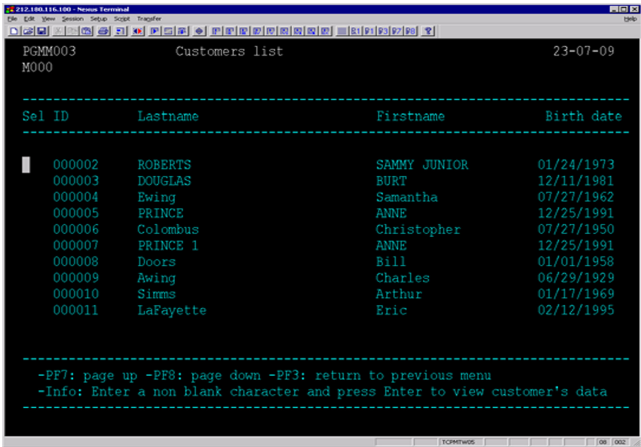
-Lastname.....: DOUGLAS
-Firstname.....: BURT
-Address.....: 12, COLLINGWOODS DR
-City.....: MARIETTA
-State.....: GA
-Birth date....: 12/11/1981 (mm/dd/yyyy)
-Email.....: BIGDOUGAFRIENDS.NET
-Phone number..: 5556531100

-----
-PF03: return to previous screen
-----
Datan OK. Press PF12 to confirm

```

# Printing Customer Reports

- 1. From the `main` menu, press PF04 to access the Customer List first page.

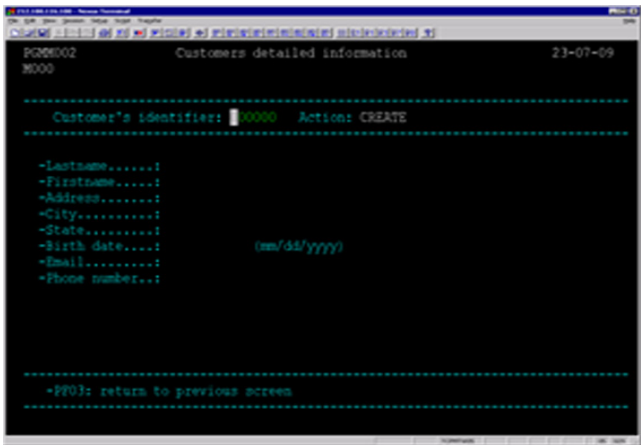


- 2. Press PF08 to display the next page in the list. Press PF07 to display a previous page in the list.

**Note:** You can perform a customer inquiry by entering any non-blank character in the selection (Sel) column to the left of each customer.

# Adding New Customers

- 1. From the `main` menu press PF05, the following screen is displayed.



- Action is set to CREATE in the screen header and fields are unprotected.
2. Enter the new customer data including a valid non-existing identifier, and press ENTER.  
The program validates the record; when no errors are found, a confirmation message is displayed.
  3. Press PF12 to create a new customer record.  
A New customer added message is displayed.
  4. Press PF03 to return to the main menu.

## Simple App APIs

This section lists the CICS APIs which are used within the Simple App application programs.

### MAP APIs

#### SEND

```
EXEC CICS
    SEND MAP('MAPM000') MAPSET('MAPM000') ERASE
END-EXEC.

EXEC CICS
    SEND MAP('MAPM000') MAPSET('MAPM000') CURSOR ERASE
END-EXEC.

EXEC CICS
    SEND MAP('MAPM001') MAPSET('MAPM001') FROM(MAPM001O) ERASE
END-EXEC.
```

#### RECEIVE

```
EXEC CICS
    RECEIVE MAP('MAPM000') MAPSET('MAPM000')
END-EXEC.

EXEC CICS
```

```
        RECEIVE MAP('ORDMAP1') MAPSET('ORDSET1') INTO(ORDMAP1)
END-EXEC.
```

## NAVIGATION APIs

### RETURN

```
EXEC CICS
    RETURN
END-EXEC.

EXEC CICS
    RETURN TRANSID('M000') COMMAREA(COMM-RECORD)
                                LENGTH(LENGTH OF COMM-RECORD)
END-EXEC.
```

### XCTL

```
EXEC CICS
    XCTL PROGRAM(PGM-DEST) COMMAREA(COMM-RECORD)
                                LENGTH(LENGTH OF COMM-RECORD)
END-EXEC.
```

### ABEND

```
EXEC CICS
    ABEND ABCODE('META')
END-EXEC.
```

## VSAM APIs

### STARTBR

```
EXEC CICS STARTBR DATASET ('ODCSF0')
                                RIDFILD (CUST-FILE-KEY)
```



```

                                EQUAL
                                RESP      (RESPONSE-CODE)

END-EXEC.

EXEC CICS STARTBR DATASET ('ODCSF0')

                                RIDFILD (CUST-FILE-KEY)

                                GTEQ

                                RESP      (RESPONSE-CODE)

END-EXEC.

```

## READ

```

EXEC CICS

    READ FILE ('ODCSF0')

        INTO (VS-ODCSF0-RECORD)

        RIDFLD (CUST-FILE-KEY)

        LENGTH (LENGTH OF VS-ODCSF0-RECORD)

        EQUAL

        RESP (RESPONSE-CODE)

END-EXEC.

EXEC CICS READ DATASET ('ODCSF0')

        INTO      (READ-ODCSF0-RECORD)

        LENGTH    (LENGTH OF READ-ODCSF0-RECORD)

        RIDFLD    (CUST-FILE-KEY)

        EQUAL

        UPDATE

        RESP      (RESPONSE-CODE)

END-EXEC.

```

## WRITE

```
EXEC CICS WRITE DATASET  ('ODCSF0')  
  
          FROM      (MAJ-ODCSF0-RECORD)  
  
          LENGTH    (LENGTH OF MAJ-ODCSF0-RECORD)  
  
          RIDFLD     (MAJ-CUSTIDENT)  
  
          KEYLENGTH  (6)  
  
          RESP       (RESPONSE-CODE)  
  
END-EXEC.
```

## REWRITE

```
EXEC CICS REWRITE DATASET ('ODCSF0')  
  
          FROM      (READ-ODCSF0-RECORD)  
  
          LENGTH    (LENGTH OF READ-ODCSF0-RECORD)  
  
          RESP      (RESPONSE-CODE)  
  
END-EXEC.
```

## READNEXT

```
EXEC CICS READNEXT DATASET ('ODCSF0')  
  
          INTO      (CLT-ODCSF0-RECORD)  
  
          LENGTH    (LENGTH OF CLT-ODCSF0-RECORD)  
  
          RIDFLD     (CUST-FILE-KEY)  
  
          RESP       (RESPONSE-CODE)  
  
END-EXEC.
```

## READPREV

```
EXEC CICS READPREV DATASET ('ODCSF0')  
  
          INTO      (CLT-ODCSF0-RECORD)  
  
          LENGTH    (LENGTH OF CLT-ODCSF0-RECORD)  
  
          RIDFLD     (CUST-FILE-KEY)
```

```

                                RESP      (RESPONSE-CODE)

END-EXEC.

```

## ENDBR

```

EXEC CICS ENDBR DATASET ('ODCSF0')

END-EXEC.

```

## DELETE

```

EXEC CICS DELETE FILE      ('ODCSF0')

                                RIDFLD    (CUST-FILE-KEY)

                                RESP      (RESPONSE-CODE)

END-EXEC.

```

# MISCELLANEOUS APIs

## TIME

```

EXEC CICS

                                ASKTIME ABTIME (ABS-TIME)

END-EXEC.

EXEC CICS

                                FORMATTIME ABTIME (ABS-TIME)

                                DDMMYY (VDATEO) DATESEP ('-')

END-EXEC.

```

## HANDLE CONDITIONS

```

EXEC CICS

                                IGNORE CONDITION      MAPFAIL

END-EXEC.

EXEC CICS

                                HANDLE CONDITION PGMIDERR (PGM-NOTFOUND)

```

END-EXEC .

## Simple App Documentation References

You may find a detailed description of the CICS APIs at the following address:

[http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp?topic=/com.ibm.cics.ts31.doc/dfhp4/topics/dfhp4\\_commands.htm](http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp?topic=/com.ibm.cics.ts31.doc/dfhp4/topics/dfhp4_commands.htm)

## Detailed Description of the CICS APIs

The following CICS APIs are used in the Simple App application.

**Table 3-2 Simple App CICS APIs**

API	Description
ASKTIME	Request current date and time of day.
DELETE	Delete a record from a file - VSAM KSDS, VSAM RRDS, and data tables only.
ENDBR	End browse of a file.
FORMATTIME	Transform absolute date and time into a specified format.
LINK	Link to another program expecting return. The external CICS interface (EXCI) provides a LINK command that performs all six commands of the interface in one invocation. See the CICS External Interfaces Guide for information about the EXCI.
READ	Read a record from a file.
READNEXT	Read next record during a browse of a file.
READPREV	Read previous record during a file browse; VSAM and data tables only.
RECEIVE MAP	Receive screen input into an application data area. For further information about BMS, see the
RETURN	Return program control.
REWRITE	Update a record in a file.
SEND MAP	Send mapped output data to a terminal. The keywords are separated into those supported by minimum, standard, and full BMS. For further information about BMS, see the CICS Application Programming Guide.

**Table 3-2 Simple App CICS APIs**

API	Description
START	Start task at a specified time.
STARTBR	Start browse of a file.
WRITE	Write a record.
XCTL	Transfer program control.

This document is based upon the following reference guide: [CICS Transaction Server for z/OS Application Programming Reference](#)

## The Simple App Application