

# ACME Phase 2(c) RGA Enabling External User Functionality

Version 6.1

Alvin J. Alexander  
<http://www.devdaily.com>

Jan. 28, 2004

Copyright DevDaily Interactive (devdaily.com), 2004.  
All Rights Reserved.

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Assumptions</b>	<b>9</b>
2.0.1	Application Mode Assumptions . . . . .	10
2.0.2	User Interface Assumptions . . . . .	10
2.0.3	Documentation, Help System, and Graphics Assumptions . . . . .	11
2.0.4	<i>Database Assumptions</i> . . . . .	11
2.0.5	Database Service Assumptions . . . . .	12
2.0.6	Testing Assumptions . . . . .	12
<b>3</b>	<b>Requirements</b>	<b>14</b>
3.1	External User Login and Access Control Requirements . . . . .	14
3.1.1	Access Control Requirements, General Behavior . . . . .	14
3.1.2	Access Control Requirements, Security <i>is not</i> Enabled . . . . .	15
3.1.3	Access Control Requirements, Security <i>is</i> Enabled . . . . .	16
3.1.4	Initial database information . . . . .	19
3.1.5	External user logins . . . . .	19
3.2	<i>Database Requirements</i> . . . . .	20
3.2.1	<i>Database Selection</i> . . . . .	20
3.3	<i>Documentation Notes, Errata</i> . . . . .	20
<b>4</b>	<b>Use Cases</b>	<b>22</b>
4.1	External User Login . . . . .	24
4.1.1	Notes . . . . .	24
4.1.2	Actors . . . . .	24
4.1.3	Pre-Conditions . . . . .	24
4.1.4	Access Control <i>is not</i> Enabled . . . . .	24
4.1.5	Alternative Path #1, Access Control <i>is</i> Enabled . . . . .	25

4.1.6	Alternative Path #2, Security Enabled, Administrator Logs In . . . . .	27
4.1.7	Alternative Path #3, <i>User Selects a Different Database During Login</i> . . . . .	28
4.1.8	"Used" Use Cases . . . . .	28
4.1.9	Other Requirements . . . . .	29
4.2	Add User Account . . . . .	30
4.2.1	Notes . . . . .	30
4.2.2	Actors . . . . .	30
4.2.3	Pre-Conditions . . . . .	30
4.2.4	Basic Path: Create a New User Account . . . . .	30
4.2.5	Post-Conditions . . . . .	31
4.2.6	"Used" Use Cases . . . . .	31
4.2.7	Other Requirements . . . . .	31
4.3	List User Accounts . . . . .	32
4.3.1	Notes . . . . .	32
4.3.2	Actors . . . . .	32
4.3.3	Pre-Conditions . . . . .	32
4.3.4	Flow of Events: Administrator . . . . .	32
4.3.5	Flow of Events: Non-Administrator . . . . .	32
4.3.6	"Used" Use Cases . . . . .	33
4.3.7	Other Requirements . . . . .	33
4.4	Edit User Account . . . . .	34
4.4.1	Notes . . . . .	34
4.4.2	Actors . . . . .	34
4.4.3	Pre-Conditions . . . . .	34
4.4.4	Basic Path: Administrator Edits User Account . . . . .	34
4.4.5	Alternative Path #2, External User Changes Their Account Information . . . . .	35
4.4.6	Post-Conditions . . . . .	36
4.4.7	Other Requirements . . . . .	36
4.5	Delete a User Account . . . . .	37
4.5.1	Notes . . . . .	37
4.5.2	Actors . . . . .	37
4.5.3	Pre-Conditions . . . . .	37
4.5.4	Basic Path: Administrator Deletes One User Account . . . . .	37
4.5.5	Alternate Path #1: Administrator Deletes More Than One User Account . . . . .	38
4.5.6	"Used" Use Cases . . . . .	38
4.5.7	Other Requirements . . . . .	38

4.6	Add Group Account . . . . .	40
4.6.1	Notes . . . . .	40
4.6.2	Actors . . . . .	40
4.6.3	Pre-Conditions . . . . .	40
4.6.4	Basic Path: Administrator Adds a Group . . . . .	40
4.6.5	"Used" Use Cases . . . . .	41
4.6.6	Other Requirements . . . . .	41
4.7	List Groups . . . . .	42
4.7.1	Notes . . . . .	42
4.7.2	Actors . . . . .	42
4.7.3	Pre-Conditions . . . . .	42
4.7.4	Basic Path: User Displays a List of Groups . . . . .	42
4.7.5	"Used" Use Cases . . . . .	43
4.7.6	Other Requirements . . . . .	43
4.8	Edit a Group . . . . .	44
4.8.1	Notes . . . . .	44
4.8.2	Actors . . . . .	44
4.8.3	Pre-Conditions . . . . .	44
4.8.4	Basic Path: Administrator Edits a Group . . . . .	44
4.8.5	Alternate Path #1: Non-Administrator Views Group Information . . . . .	45
4.8.6	"Used" Use Cases . . . . .	45
4.8.7	Other Requirements . . . . .	45
4.9	Delete a Group . . . . .	46
4.9.1	Notes . . . . .	46
4.9.2	Actors . . . . .	46
4.9.3	Pre-Conditions . . . . .	46
4.9.4	Basic Path: Administrator Deletes One Group . . . . .	46
4.9.5	Post-Conditions . . . . .	46
4.9.6	"Used" Use Cases . . . . .	46
4.9.7	Other Requirements . . . . .	47
4.10	Enable/Disable Security . . . . .	48
4.10.1	Notes . . . . .	48
4.10.2	Actors . . . . .	48
4.10.3	Pre-Conditions . . . . .	48
4.10.4	Basic Path: Enabling Security . . . . .	48
4.10.5	Basic Path: Disabling Security . . . . .	48
4.10.6	"Used" Use Cases . . . . .	49
4.10.7	Other Requirements . . . . .	49
4.11	Become Administrator . . . . .	50

## Contents

---

4.11.1	Notes . . . . .	50
4.11.2	Actors . . . . .	50
4.11.3	Pre-Conditions . . . . .	50
4.11.4	Basic Path: Become Administrator . . . . .	50
4.11.5	"Used" Use Cases . . . . .	51
4.11.6	Other Requirements . . . . .	52
4.12	Creating a New ACME Job . . . . .	53
4.12.1	Notes . . . . .	53
4.12.2	Actors . . . . .	53
4.12.3	Priority . . . . .	53
4.12.4	Status . . . . .	53
4.12.5	Pre-Conditions . . . . .	53
4.12.6	Basic Path: No job is currently open in the ACME Editor . . . . .	53
4.12.7	Alternative Path #1, Already a "clean" job open in the ACME Editor . . . . .	54
4.12.8	Alternative Path #2, Already a "dirty" job open in the ACME Editor . . . . .	55
4.12.9	Sub Use Case - Prompt User for Job Information . . . . .	56
4.12.10	"Used" Use Cases . . . . .	56
4.12.11	Other Requirements . . . . .	56
4.13	Search/List ACME Jobs . . . . .	58
4.13.1	Notes . . . . .	58
4.13.2	Actors . . . . .	58
4.13.3	Pre-Conditions . . . . .	58
4.13.4	Non-Administrator Selects the Search/List Option . . . . .	58
4.13.5	"Used" Use Cases . . . . .	59
4.13.6	Other Requirements . . . . .	60
4.14	Open a ACME Job . . . . .	61
4.14.1	Notes . . . . .	61
4.14.2	Actors . . . . .	61
4.14.3	Pre-Conditions . . . . .	61
4.14.4	Basic Path - No job is open in the ACME Editor . . . . .	61
4.14.5	Alternative Path #1, There is already a "clean" job open in the ACME Editor . . . . .	63
4.14.6	Alternative Path #2, Already a "dirty" job open in the ACME Editor . . . . .	64
4.14.7	Sub Use Case: Open Job in New Editor . . . . .	65
4.14.8	"Used" Use Cases . . . . .	65
4.14.9	Other Requirements . . . . .	66

## Contents

---

4.14.10	Issues . . . . .	67
4.15	Save a ACME Job . . . . .	68
4.15.1	Notes . . . . .	68
4.15.2	Actors . . . . .	68
4.15.3	Pre-Conditions . . . . .	68
4.15.4	Basic Path: User Saves Job . . . . .	68
4.15.5	Alternative Path #1, The Save As ... option . . . . .	69
4.15.6	"Used" Use Cases . . . . .	70
4.15.7	Other Requirements . . . . .	70
4.15.8	Issues . . . . .	70
4.16	Delete a Job . . . . .	71
4.16.1	Notes . . . . .	71
4.16.2	Actors . . . . .	71
4.16.3	Pre-Conditions . . . . .	71
4.16.4	Basic Path - User deletes one job . . . . .	71
4.16.5	Alternative Path #1, User selects multiple jobs to delete . . . . .	72
4.16.6	"Used" Use Cases . . . . .	73
4.16.7	Other Requirements . . . . .	73
4.17	Clone a ACME Job . . . . .	74
4.18	Clear Job Lock . . . . .	75
4.18.1	Notes . . . . .	75
4.18.2	Actors . . . . .	75
4.18.3	Pre-Conditions . . . . .	75
4.18.4	Basic Path - Administrator Clears a Job Lock . . . . .	75
4.18.5	Alternate Path #1 - Non-Administrator Clears a Job Lock . . . . .	76
4.18.6	"Used" Use Cases . . . . .	76
4.18.7	Other Requirements . . . . .	76
4.18.8	Issues . . . . .	76
4.19	Exiting the Application . . . . .	77
4.19.1	Notes . . . . .	77
4.19.2	Actors . . . . .	77
4.19.3	Pre-Conditions . . . . .	77
4.19.4	Basic Path: One Editor Frame is Open . . . . .	77
4.19.5	"Used" Use Cases . . . . .	78
4.19.6	Other Requirements . . . . .	78
4.20	Add a ACME Database . . . . .	79
4.20.1	Notes . . . . .	79
4.20.2	Actors . . . . .	79
4.20.3	Pre-Conditions . . . . .	79

## Contents

---

4.20.4	Basic Path . . . . .	79
4.20.5	"Used" Use Cases . . . . .	80
4.20.6	Other Requirements . . . . .	80
4.21	Delete Database Alias . . . . .	81
4.21.1	Notes . . . . .	81
4.21.2	Actors . . . . .	81
4.21.3	Pre-Conditions . . . . .	81
4.21.4	Basic Path . . . . .	81
4.21.5	"Used" Use Cases . . . . .	81
4.21.6	Other Requirements . . . . .	81
4.22	Edit a ACME Database Alias . . . . .	82
4.22.1	Notes . . . . .	82
4.22.2	Actors . . . . .	82
4.22.3	Pre-Conditions . . . . .	82
4.22.4	Basic Path . . . . .	82
4.22.5	"Used" Use Cases . . . . .	82
4.22.6	Other Requirements . . . . .	82
4.23	List Known Database Aliases . . . . .	83
4.23.1	Notes . . . . .	83
4.23.2	Actors . . . . .	83
4.23.3	Pre-Conditions . . . . .	83
4.23.4	Basic Path . . . . .	83
4.23.5	"Used" Use Cases . . . . .	83
4.23.6	Other Requirements . . . . .	83
<b>5</b>	<b>Database Design</b>	<b>85</b>
<b>6</b>	<b>User Interface</b>	<b>87</b>
6.0.7	Menu Design . . . . .	87
6.0.8	Prototypes . . . . .	87
<b>7</b>	<b>Issues</b>	<b>88</b>
<b>8</b>	<b>Risks</b>	<b>90</b>

# Chapter 1

## Introduction

This is the Requirements Gathering and Analysis (RGA) document for the project of re-enabling the External User functionality of the ACME application.

This printed document includes sections on:

1. Assumptions
2. Requirements
3. Use Cases
4. Issues
5. Risks
6. Data

Other printed documents that should be distributed with this document to compose the full set of requirements include the following:

1. Application Prototypes
2. User Interface Storyboards

Quick dictionary:

1. CRUD - Created, read, update, delete
2. ACL - Access control list

Font information:

1. Anything listed in a **green font** is new since the release of Version 5
2. Anything listed in a **red font** is new since the release of Version 6



---

3. Anything listed in a gray font is planned for deletion

***Bold and italics phrases*** are also new since the first release of Version 5. If an entire section is new, the section may not be in an entirely red font, but there will be an indicator that the section is new.

# Chapter 2

## Assumptions

This chapter lists the known assumptions regarding this phase of development.

1. Generally speaking, we are not necessarily trying to create new functionality for the External User at this time. Rather, we are trying to further define the functionality originally outlined during Phase 1, and resolve questions that have arisen from that original outline before programming continues.
2. This application is not being shipped to any outside users in Phase 2c.
  - (a) Therefore, within Phase 2c, it will continue to be deployed through the Java WebStart tool, and no time will be spent trying to "package" the application with a different deployment tool (such as InstallShield or similar).
  - (b) **Therefore, this phase concludes when user acceptance testing has been completed.**
3. Job types
  - (a) Development during this sub-phase is only for Unplanned job types
4. **Only Consultant developers will be working on the Java code during this phase.**
5. **The system will run on the following computer platforms:**
  - (a) **Windows 2000**
  - (b) **Windows XP**
  - (c) **Windows NT**
  - (d) **Mac - OS/X 10.2 and newer**

---

### 2.0.1 Application Mode Assumptions

1. The application shall work in two modes:
2. External User Standalone
  - (a) This is a single user, non-networked system
  - (b) The database will reside on the user's workstation
3. External User Network (Client-Server)
  - (a) This is a client instance of ACME
  - (b) The client may be running on a laptop or desktop
  - (c) The database may be on this user's workstation, or on another workstation or server

This section is new, but the line items have all been reviewed/approved

### 2.0.2 User Interface Assumptions

1. General
  - (a) Java user interface standards will be followed unless otherwise specified
2. Title bars
  - (a) Accurate and consistent titles will be displayed in all windows, frames, and dialogs
3. Dialogs
  - (a) The user will be able to cancel all dialogs by hitting the [Esc] key
  - (b) Disposing a dialog will have the same effect as hitting [Cancel]
4. Text fields
  - (a) Text fields will match the corresponding database fields.
  - (b) Text fields will not allow users to type invalid characters
    - i. For instance, if a user tries to type alphabetic characters into a numeric field, the alpha characters will not be displayed
    - ii. This was marked as an issue
5. Textareas
  - (a) Tab keys will not be allowed in text areas
    - i. Tab keys will specifically move the user out of the textarea
6. Tables
  - (a) Tables will not allow columns to be moved around.
  - (b) Tables will not allow columns sorting unless otherwise indicated.

This is a new section

- 
7. Input focus
    - (a) Focus on input widgets will follow the standard left to right, top to bottom flow unless otherwise specified.
  8. Default button
    - (a) A default button will be specified on screens, and will logically follow user focus.
      - i. For example, on a wizard requiring multiple dialogs, the "Next" button on each dialog will be the default button.

### 2.0.3 Documentation, Help System, and Graphics Assumptions

1. No Help system is going to be included in the application during Phase 2c
2. There will be no reports (hardcopy) that users can print during Phase 2c
3. No form of documentation is offered beyond Intranet (Wiki) documentation, similar to what was created during Phase 1
  - (a) This is primarily because no form of documentation has been agreed upon at this time
4. The only graphics art work for this phase is:
  - (a) Work to add a "database" icon to the splash screen
  - (b) Graphics work for the Job/Open wizard
5. Three person-days of documentation effort will be provided by Consultant. Documentation format is TBD.

This is a new section

### 2.0.4 Database Assumptions

1. ACME ID's in the External ACME database are the same as the ACME\_id on the AS/400
2. ACME will develop software and processes to generate SQL statements to populate all necessary database tables
  - (a) The database data that is generated and shipped with the ACME client will be the same for all users
    - i. For instance, Ralph's ACME client database will initially contain the exact same data as Margaret's client database

- 
- ii. Multiple database servers can be set up with different ACME databases
  - (b) Note that ACME cannot generate the binary "database" itself, but can generate the necessary CREATE and INSERT statements necessary to populate the database
  - (c) Consultant will run the ACME scripts to create the necessary database tables
  - (d) While it is expected that ACME will create this software and processes, Consultant will assist ACME in understanding the required syntax
  - (e) There will be no screens created or other code written to accommodate the process of copying or moving a job from one database to another during Phase 2c.
3. The ability to define database connections is encompassed in the screens that the user can see during application startup
- (a) There are no other database menus or options in ACME at this time

### 2.0.5 Database Service Assumptions

- 1. In the long run, the database is going to need to be installed as a service on both Windows and Mac systems. However, this requires an installer program to set up properly, and we are not going to be using an installer until Phase 2E.
- 2. For Phase 2C, the "database service" must be started by having the user start a batch file to simulate having a database service.
  - (a) The user will be able to stop this service by typing "Ctrl-C" in the Command window on Windows PCs
  - (b) The user will be able to stop the service by killing the process on Mac computers
- 3. Phase 2C is being estimated with the assumption that this is an acceptable solution (until we use an installer in Phase 2E)
- 4. SHLNXTTEST1 will be configured as a network database server, and users will be able to set up and use connections to it

This is a new section

### 2.0.6 Testing Assumptions

- 1. Testing and acceptance of the functionality is the full responsibility of ACME.

New section and line items

- 
2. ACME testing will be required periodically during application development.
  3. ACME final testing and acceptance will be completed within two weeks of the final delivery date, assuming that ACME is not waiting for Consultant at this time.
  4. Consultant can assist in the creation of a test plan if desired, but that time will not be included in our bid.

# Chapter 3

## Requirements

Requirements for this application that cannot be easily specified in the Use Cases are listed in this section.

### 3.1 External User Login and Access Control Requirements

For users external to PPC, the behavior of the Access Control System is defined in the following sections.

#### 3.1.1 Access Control Requirements, General Behavior

This section describes the general behavior of the access control system, regardless of whether access control is enabled, or not.

1. For external users, enabling of "access control" is optional
2. The access control system can be enabled or disabled at any time, including:
  - (a) Initial application installation
  - (b) Later application use
3. Access control applies to all users
  - (a) If enabled, all users will be limited by their group assignments
  - (b) If disabled, all users have free access to system resources
4. Initial user accounts
  - (a) ACME will ship with two initial user accounts:

### 3.1. External User Login and Access Control Requirements

---

- i. Administrator
  - ii. Unknown User
- (b) These two initial user accounts cannot be deleted
- 5. Administrator account
  - (a) The Administrator user is in a group also named Administrator
  - (b) Regardless of whether security is enabled, there will always be an administrative user
  - (c) The administrator of external security will have the username "Administrator"
  - (d) **The password for this account is initially "admin"**
  - (e) **A password for this account is mandatory**
  - (f) The password for this account can be changed only by the Administrator
  - (g) The Administrator user account cannot be deleted or edited
  - (h) The Administrator group cannot be deleted or edited
- 6. Unknown User account
  - (a) The Unknown User user is in a group named Unknown User
  - (b) The Unknown User user cannot be deleted or edited
  - (c) The Unknown Group group cannot be deleted or edited
  - (d) The Unknown Group group has full access to all functional areas other than User and Group management
  - (e) Users cannot log in as the user Unknown User when security is enabled
- 7. CRUDing User Accounts
  - (a) The Administrator is the only user that can add, delete, and edit user account information
  - (b) All users can view usernames, group names, group members, and group rights without restriction.
- 8. The Administrator can log in and use ACME, and has no security restrictions whatsoever.

#### 3.1.2 Access Control Requirements, Security *is not* Enabled

This section describes the specific behavior of the access control system under the condition that the security features of the system *are disabled*.

- 1. *The login screen is not displayed*



### 3.1. External User Login and Access Control Requirements

---

2. All users will have unlimited read/write access to all ACME functions *except* User and Group management
3. ***The Administrator can only log in through the "Become Administrator" process***
  - (a) Any User or Group changes will not take effect until security is enabled

#### 3.1.3 Access Control Requirements, Security is Enabled

This section describes the specific behavior of the access control system under the condition that the security features of the system have been enabled.

1. External Access Control Levels (ACL) will be based on the same functional areas that the internal security is based on
  - (a) For example, a user may or may not have the right to edit an invoice
2. External ACL will be controlled by the Administrator account
  - (a) The Administrator can:
    - i. Add, edit and remove groups
    - ii. Add, edit and remove user accounts
    - iii. Assign functional areas to groups
    - iv. Add, edit and remove user-to-group relationships
3. **If multiple users are going to log in, security must be enabled**
  - (a) Note: We do not want to program for this until Phase 2E when an actual installation program will be used

#### Functional areas

**This section is completely new.**

1. **Functional areas for Phase 2C are defined as follows:**
  - (a) Job - New
    - i. User can either create new jobs, or not
  - (b) Job - Edit
    - i. User can either edit jobs, or they are read-only
  - (c) Job - Delete
    - i. User can either delete jobs or not

### 3.1. External User Login and Access Control Requirements

---

- (d) Job - Clear Locks
  - (e) Become Administrator
  - (f) Switch Databases
  - (g) Job - Save As
    - i. The ability to create a job implies the ability to use "Save As..."
    - ii. The ability to delete a job implies the ability to overwrite an existing job
  - (h) Job - List Jobs
    - i. Issue: Is there any need to restrict this?
  - (i) The following functional areas belong only to the Administrator
    - i. User - New
    - ii. User - Edit
    - iii. User - Delete
    - iv. Group - New
    - v. Group - Edit
    - vi. Group - Delete
  - (j) List User Accounts
2. The following functional areas are specifically not being created. It is assumed that all users can CRUD database connections.
- (a) Database - New
  - (b) Database - Edit
  - (c) Database - Delete
3. **This list of functional areas is the same for all customers**
- (a) i.e., all customers will have "User - New", "User - Edit"
  - (b) i.e, Bobit will not have one list of functional areas, Medical World another list, etc.
4. There are three possible access levels for each functional area:
- (a) Hidden
    - i. Generally speaking, menu options will not be visible
  - (b) View
    - i. Generally speaking, menu options will be visible, but not enabled
  - (c) Edit
    - i. Menu options will be enabled

### 3.1. External User Login and Access Control Requirements

---

5. One and only one access level can be specified per functional area, per group created
  - (a) Ex: The group READ\_ONLY is defined to have one access level of "VIEW" for the "Edit Job" functional area.

#### External User Groups

When access control is enabled, User Groups can be created by the Administrator to represent user roles.

1. External User Groups
  - (a) Functional areas will be assigned to groups
  - (b) Groups cannot contain other groups
    - i. i.e., there is not a recursive nesting of one group inside of another group, inside of another ...
    - ii. Example:
      - A. One group is named "Order Entry"
      - B. A second group is named "Book Map"
      - C. Order Entry cannot contain Book Map
      - D. Book Map cannot contain Order Entry
  - (c) Group name restrictions
    - i. Group names can be up to 32 characters
    - ii. Group names can contain the following characters:
      - A. A-Z, a-z, 0-9, \_, -, and blank spaces
      - B. The name must begin with a character or number
2. ACME will ship with one default Group named something like "ALL\_RIGHTS"
  - (a) This group will have read/write access for all functional areas

#### External Users

With access control enabled, the Administrator must create user accounts to let users log in.

1. External Users
  - (a) Users must login when starting the application
  - (b) Users will be able to change their passwords
  - (c) Users can belong to only one group
  - (d) User name restrictions

### 3.1. External User Login and Access Control Requirements

---

- i. User names can be up to 20 characters
- ii. User names can contain the following characters:
  - A. A-Z, a-z, 0-9, \_, -, and blank spaces
  - B. The name must begin with a character or number

#### External User Passwords

- 1. A User account must always have an associated password
- 2. Passwords
  - (a) Password restrictions
    - i. Passwords can be up to 20 characters
    - ii. Passwords can contain any ASCII character
- 1. Password changing
  - (a) Users can change their own passwords
  - (b) The Administrator can change any user's password

#### 3.1.4 Initial database information

- 1. The User database will ship with two accounts:
  - (a) Administrator
  - (b) Unknown User
- 2. The Group database will ship with two accounts:
  - (a) Administrator
  - (b) Unknown Group
- 3. The Customer database will ship with one or more PPC customer accounts

#### 3.1.5 External user logins

- 1. Any user can log into the system more than once simultaneously
  - (a) This includes Administrators
  - (b) This includes multiple logins per one machine
  - (c) This includes multiple logins from multiple workstations

### 3.2 *Database Requirements*

1. We will simulate the process of shipping a database with customers by including a database with ACME that has multiple customers
2. The first time the user downloads ACME they will have only a local database
  - (a) This initial database will be the same for all users
3. For all subsequent downloads of the ACME application, the original database will remain in the state it was in prior to the download
  - (a) For instance, if a user created a job, that job will still remain in the database
  - (b) The exception to this requirement is when a database change occurs. During that time, it may be necessary to download an entirely new database, thereby destroying any changes a user had made before the download.
4. During Phase 2c, to create the Client/Server environment, users will be able to
  - (a) Access a database server running on SHLNXTEST1
  - (b) Access databases on other user's PCs
    - i. This statement assumes that the "enable multiuser mode" requirements are approved

#### 3.2.1 *Database Selection*

1. During startup, ACME will always start by trying to connect to the last database the user accessed
2. However, users will be able to select a database to use as the ACME applications starts up
  - (a) For instance, the user may click on icon on the splash screen during application startup to display a "database selection" screen
3. The name of the database the user is using will always be displayed in ACME's status bar

### 3.3 *Documentation Notes, Errata*

The following general purpose guidelines should be used to override potential conflicts in this documentation. If documentation appears to be out of sync, use these general rules to resolve conflicts, if possible.

New section  
and line  
items

### 3.3. *Documentation Notes, Errata*

---

1. In all cases throughout this documentation, if a username "nobody" has not been changed, "nobody" should have been changed to "Unknown User"
2. In all cases, if a user\_id of 0 or 1 is referred to, this was intended to refer to the users Administrator and Unknown User, respectively
3. If multiple users are going to access a database, then user accounts are required, which implies that security must be enabled
  - (a) The definition has been changed so that the system is now either in (a) single-user mode, or (b) multi-user mode.
  - (b) If the system is in multiuser mode, security is required, and user accounts are required.
  - (c) If the system is in single user mode, security is turned off, and users do not see the login window.
4. Administrators can only manage user accounts, group accounts, and clear job locks. They specifically cannot edit jobs.
5. Users can have multiple ACME Editors ("sessions") open at one time on a PC.
  - (a) Each session can be connected to a different database, if desired.
6. There is no longer a concept of an "inactive" user account. Any references to an inactive user account, deactivating users, or reactivating users are incorrect.

## Chapter 4

# Use Cases

Use Cases define the processes that the application will provide to satisfy the needs of the actors that will use this application.

Each use case is defined in a step-by-step listing of what the user will see and do while using that particular portion of the ACME application.

---

This is a list of the use cases that you will find on the following pages:

1. User Login (also referred to as ACME Startup)
2. User - Add
3. User - List
4. User - Edit
5. User - Delete
6. Group - Add
7. Group - List
8. Group - Edit
9. Group - Delete
10. Become Administrator
11. Job - New
12. Job - List
13. Job - Open
14. Job - Save
15. Job - Delete
16. Job Clear Lock
17. ACME - Exit
18. Database - Add
19. Database - List
20. Database - Delete
21. Database - Edit



## 4.1 External User Login

This use case defines the process of logging the external user into the ACME application.

### 4.1.1 Notes

1. All user types can log in more than once
  - (a) They can log in at multiple workstations
  - (b) They can have multiple ACME Editors open on their desktop (each with a different job)
2. *Users can access multiple databases*
  - (a) One ACME Editor can be connected to a local database, while another is connected to a remote database
  - (b) The "default" database is the last database the **current** user accessed from the current workstation

ISSUE  
HERE

### 4.1.2 Actors

1. External User
2. Administrator

### 4.1.3 Pre-Conditions

1. ACME is properly installed on the user's system
2. Access control is either disabled or enabled
3. If access control is enabled on the system:
  - (a) A user account may or may not be available for the user attempting to log in

### 4.1.4 Access Control *is not* Enabled

1. The user starts ACME
  - (a) Note - the system *does not* display a login screen
2. The system connects to the "**default**" database
3. The system logs the user in as the user "Unknown User"
4. The system grants all access rights to the user, *except User and Group administration rights*
5. The system displays the ACME Editor

#### 4.1. External User Login

---

- (a) The ACME Navigator is not displayed
  - (b) Menu options including Job/Open, Job/New, Job/Exit, and Help are available
  - (c) No jobs are opened by default
6. A "Job Open/Create" dialog/wizard is displayed to the user that lets them follow four courses of action
- (a) Start the process of creating a new job
  - (b) Open a recent job
    - i. The system will display the most recent jobs (10 max), in sorted order showing the most recent first
    - ii. Note: This list will come from a data file maintained by ACME under the user's home directory on their computer
  - (c) Open an older job by starting the List Jobs use case
  - (d) Close this dialog

#### Post-Conditions

1. ACME is started
2. The user is connected to the "*default*" database
3. The user is logged in as the user "Unknown User"
4. Full system access is granted
  - (a) Note - this means the user can CRUD any job
  - (b) Note - this means the user can CRUD any job details
5. No jobs are loaded into the Editor
6. The ACME Navigator is not displayed
7. Certain menu options are enabled (Job and Help-related)
8. **Recent file list**
  - (a) When the user logs in the system will display a list of the user's most recently worked on 10 jobs
  - (b) **Change:** The recent file list will appear only under the Job menu, in a manner similar to Microsoft Word

#### 4.1.5 Alternative Path #1, Access Control *is* Enabled

1. The user starts ACME
2. The system connects to the "default" database
3. The system displays the ACME login screen

#### 4.1. External User Login

---

4. *If the user cancels the operation (such as the Cancel button or [Esc] key) the login is aborted and the application shuts down*
5. The user enters a name and password
6. The system validates the username and password against the access control database
7. If the username/password combination is *invalid*:
  - (a) The system lets the user know that the username/password combination is invalid
  - (b) The login screen is re-displayed
  - (c) Assuming that this process continues to occur, the login screen is again re-displayed to the user
    - i. There is no limit to the number of times the screen is displayed
    - ii. The screen is never disabled
    - iii. The screen is always shown as quickly as possible by the system, without any "delay" for security reasons
8. If the username/password combination is valid, continue
9. If the system detects that the user is already logged in:
  - (a) The system notifies the user of this
    - i. This is an informational statement only; it does not prevent the user from logging in
10. The system retrieves the user's rights
11. If a problem is detected with the configuration of the user's rights:
  - (a) The system prompts the user to contact the Administrator
  - (b) The system tells the user that they cannot login to the system, and does not let them into the system
12. The system displays the ACME Editor
  - (a) The ACME Navigator is not displayed
  - (b) No jobs are opened by default
13. The "Job Open/Create" dialog is displayed to the user that lets them easily open or create a job

#### **Post-Conditions**

1. The user may or may not be logged into the system

#### 4.1. External User Login

---

2. The system is connected to the "default" database
3. If the user has logged into the system:
  - (a) Their access control rights are loaded into memory
4. No jobs are loaded into the ACME Editor
5. The ACME Navigator is not displayed
6. Certain menu options are enabled (Job and Help-related)
7. **Recent file list**
  - (a) When the user logs in the system will display a list of the user's most recently worked on 10 jobs
  - (b) **Change: The recent file list will appear only under the Job menu, in a manner similar to Microsoft Word**

##### 4.1.6 Alternative Path #2, Security Enabled, Administrator Logs In

1. (Security is enabled)
2. The Administrator starts ACME
3. The system connects to the "default" database
4. The system displays the login screen
5. The Administrator enters their username ("Administrator") and password
6. The system displays the ACME Editor
  - (a) The ACME Navigator is not displayed
  - (b) Menu options including Job—Open, Job—New, Job—Exit, and Help are available
  - (c) The menu options to edit User and Group accounts are also enabled
  - (d) No jobs are opened
7. The Job Open/Create dialog is *not* displayed

##### Post-Conditions

1. ACME is started
2. The system is connected to the "current" database
3. Full system access is granted
4. No jobs are loaded
5. The ACME Navigator is not displayed

#### 4.1. External User Login

---

6. Certain menu options are enabled (Job and Help-related, and User and Group management)
7. **Recent file list**
  - (a) The Recent File List does not apply to the Administrator

##### 4.1.7 *Alternative Path #3, User Selects a Different Database During Login*

1. The user starts ACME
2. During startup the user does something (such as holding down the [Ctrl] key) to let the system know that they want to choose a database other than the default database
3. The system displays a list of databases that they user can choose from
4. The user selects a database they would like to connect to
5. If the system encounters an error connecting the user to the database
  - (a) The system displays an error message to the user
  - (b) The user is returned to the "select database" screen
6. Else, the system connects the user to the desired database
7. Note - the remainder of this use case depends on whether that database has access control enabled or disabled
  - (a) The login process for these conditions should be the same as previously described for these conditions

##### Post-Conditions

1. The user is connected to a database other than the "default" database
2. The "default" database is always whatever database the user connected to during the last ACME startup process **on the current computer**
3. **Recent file list**
  - (a) When the user logs in the system will display a list of the user's most recently worked on 10 jobs
  - (b) This list will appear under the Job menu, in a manner similar to Microsoft Word

##### 4.1.8 "Used" Use Cases

1. **List Known Databases**

#### 4.1.9 Other Requirements

1. Regarding ACME "first-time" startup:
  - (a) System security is disabled
  - (b) The Administrator user is the only user account
  - (c) The database is populated with a list of customer accounts

## 4.2 Add User Account

This use case describes the process of adding a new user to the ACME system. (branch to Sub Use Case (Section 1a on page 24))

### 4.2.1 Notes

1. There are no restrictions on the number of user accounts that can be on the system.
2. Only the Administrator can add a new user account to the system.

### 4.2.2 Actors

1. Administrator

### 4.2.3 Pre-Conditions

1. The Administrator is logged into the system.
2. Security may or may not be enabled.

### 4.2.4 Basic Path: Create a New User Account

1. The Admin selects an option to add a new user to the system
2. The system prompts the Admin for
  - (a) the following required information:
    - i. Username
    - ii. Password (entered twice)
    - iii. The group that the user is assigned to
  - (b) and the following non-required information:
    - i. First name
    - ii. Middle Initial
    - iii. Last name
3. The Admin enters these parameters and submits the form
4. If any of the required fields are not completed, the Administrator is informed which required fields are missing
  - (a) Input focus is returned to the first missing field
5. If the username is already in use, the system informs the Administrator that they must choose a different name
6. Else, continue

## 4.2. Add User Account

---

7. If the username is invalid, the Administrator is informed of this, and valid naming criteria is displayed
8. If the password is invalid, the Administrator is informed of this, and valid naming criteria is displayed
9. The system creates the new user account

### 4.2.5 Post-Conditions

1. A new user account is created on the system with at least a username, password, and group association

### 4.2.6 "Used" Use Cases

1. List Groups
2. List Users

### 4.2.7 Other Requirements

1. When the user makes a data entry mistake and an error dialog is shown, always put input focus into the data entry field where the error occurred when the error dialog is discarded
2. Usernames are unique in the database
  - (a) For instance, the user "Kim" can only be in the database once
3. Usernames and passwords are not case sensitive
  - (a) Example: There is no difference between the usernames "Kim" and "KIM"
4. A user cannot be named "UNKNOWN USER"



## 4.3 List User Accounts

This use case describes the process of listing user accounts on the ACME system.

### 4.3.1 Notes

1. This option is available whether security is enabled or not.
2. All users can list all user accounts.
  - (a) The Administrator account is not hidden in any way.

### 4.3.2 Actors

1. Administrator
2. External user

### 4.3.3 Pre-Conditions

1. The user is logged into the system.

### 4.3.4 Flow of Events: Administrator

1. The Administrator selects an option to list the user accounts on the system
2. The system displays all user accounts
  - (a) For each account the username, and groupname fields are displayed
3. The system enables options to Add, Edit, View, and Delete user accounts

### Post-Conditions

1. The list of user accounts is displayed
2. Options are available to the Administrator to let the Admin CRUD the user account

### 4.3.5 Flow of Events: Non-Administrator

1. The User selects an option to list the user accounts on the system
2. The system displays all user accounts

### 4.3. List User Accounts

---

- (a) For each account, the username and groupname are displayed
- 3. When the user selects their own account the Edit Account option is made available to the user

#### **Post-Conditions**

- 1. A list of user accounts is displayed to the user
- 2. The **Non-Administrator user does not** have an option to view additional user account details

#### **4.3.6 "Used" Use Cases**

#### **4.3.7 Other Requirements**

## 4.4 Edit User Account

This use case describes the process of editing a user account.

### 4.4.1 Notes

1. Both of these use cases are the same regardless of whether security is enabled or disabled at the time the use case is invoked.

### 4.4.2 Actors

1. Administrator
2. External user

### 4.4.3 Pre-Conditions

1. The user is logged into the system.

### 4.4.4 Basic Path: Administrator Edits User Account

1. The Administrator lists the user accounts
2. The Administrator selects the user account they want to modify and selects an Edit User Account option
3. The system displays an "Edit User" screen, which contains the following fields in read/write mode:
  - (a) Required fields
    - i. Username
    - ii. Password(repeated twice)
    - iii. Group
  - (b) Non-required fields
    - i. First name
    - ii. Middle initial
    - iii. Last name
4. The Administrator changes all desired fields and saves the changes
5. The system validates:
  - (a) If the username has been changed
    - i. If the username is not valid, the Administrator is notified of this and the change is not allowed

#### 4.4. Edit User Account

---

- ii. If the new username is already in use, the Administrator is notified of this and the change is not allowed
  - (b) If the two passwords are not the same, the Administrator is notified of this and the change is not allowed
  - (c) If the two passwords are the same but are invalid, the Administrator is notified of this and the change is not allowed
6. The system saves the changes

#### Post-Conditions

1. Any user account field may be different
2. If the username is changed, the new username should show on any reports that show this name
  - (a) If a job was initially created by "MarthaS" and the named was later changed to "MarthaT", "MarthaT" should be displayed on future reports
  - (b) Note to developers: This implies that we are storing the "user id" with the Job, and not the user name

#### 4.4.5 Alternative Path #2, External User Changes Their Account Information

1. The user lists the user accounts
2. When the user selects their user account, the system enables the "Edit User Account" option
3. The system displays an "Edit User" screen, which contains the following fields
  - (a) Read-only mode:
    - i. Username
    - ii. Group
  - (b) Read/write mode:
    - i. First name
    - ii. Middle initial
    - iii. Last name
4. **If the user wants to change their password they select a Change Password option**
  - (a) **The system prompts the user for:**
    - i. **Their old password**

#### 4.4. Edit User Account

---

- ii. **Their new password (twice)**
- (b) The user saves their password changes
- (c) The system validates that the old password is correct
  - i. If the old password is not correct, the system tells the user that it is not correct
  - ii. After clearing the message the user is returned to the Edit User Account screen
  - iii. All password fields in the Edit User Account screen are cleared
- (d) The system validates that the two new passwords have been typed identically
  - i. If this validation fails the user is given a message that the two passwords do not match
  - ii. After clearing the message the user is returned to the Edit User Account screen
  - iii. The two "new password" fields in the Edit User Account screen are cleared
- (e) The system validates that the first of the new passwords meets length and character requirements
  - i. If the new password does not meet these requirements, the user is given a message to that effect
  - ii. After clearing the message the user is returned to the Edit User Account screen
  - iii. The two "new password" fields in the Edit User Account screen are cleared
- 5. The user changes any/all of the read/write fields, then selects the Save option
- 6. The system updates the password and name fields in the database
- 7. The system leaves the user at the List Users screen

#### 4.4.6 Post-Conditions

- 1. The user's password and/or name fields have been changed

#### 4.4.7 Other Requirements

## 4.5 Delete a User Account

This use case describes the process of deleting a user account from the system.

### 4.5.1 Notes

### 4.5.2 Actors

1. Administrator

### 4.5.3 Pre-Conditions

1. The Administrator is logged into the system.

### 4.5.4 Basic Path: Administrator Deletes One User Account

1. The Administrator selects a "List User Accounts" option
2. The Administrator selects one user from the list, and then chooses an option to delete that user account
3. If the user has any job locks the Administrator is informed "The user has the following jobs locked, and the user account cannot be deleted at this time", followed by a list of jobs and job modules that are locked
  - (a) Show the username, job short description, module, and lock [date/time](#)
  - (b) [The system offers the option of deleting all job locks at this time](#)
  - (c) [The Administrator selects the option to clear all job locks](#)
  - (d) [continue](#)
4. Else, continue
5. The system prompts with an "Are you sure?" message
6. If the Administrator replies "Yes, I am sure"
  - (a) The system deletes the user account from the database

### Post-Conditions

1. The user account is deleted from the database
2. In the future, when any jobs are listed or opened that this user created, those jobs should be displayed as being owned by Unknown User

### 4.5.5 Alternate Path #1: Administrator Deletes More Than One User Account

1. The Administrator selects a "List User Accounts" option
2. The Administrator selects two or more users from the list, and then chooses an option to delete those user accounts
3. The system prompts with a message: "You are about to delete the following user accounts, are you sure?"
4. If the Administrator replies by canceling the operation, the use case stops here
5. Else, if the Administrator replies "Yes, I am sure"
6. All user accounts that do not have jobs locked at this time are deleted
7. If any user accounts have job locks, the Administrator is informed "The following users have jobs locked. Would you like to clear their job locks and delete their user accounts? ", and their user accounts cannot be deleted at this time", followed by a list of jobs and job modules that are locked by the remaining users
  - (a) Show the username, job short description, module, and lock timestamp
  - (b) The Administrator selects the option to clear all job locks and delete the user accounts
8. The system prompts with an "Are you sure?" message
9. If the Administrator replies "Yes, I am sure"
  - (a) The system deletes the user account from the database

#### Post-Conditions

1. User accounts that did not have jobs locked are removed from the system
2. After the completion of this use case, when any jobs are listed or opened that this user created, those jobs should be displayed as being owned by Unknown User

### 4.5.6 "Used" Use Cases

1. List User Accounts

### 4.5.7 Other Requirements

1. *The Administrator account cannot be deleted*

#### 4.5. Delete a User Account

---

2. *The Unknown User account cannot be deleted*
3. Note that in this use case clearing all locks is an "all or nothing" proposition. Selectively clearing job locks is not available from this point in the application



## 4.6 Add Group Account

This use case describes the process of adding a new group to the ACME system.

### 4.6.1 Notes

1. All functional areas for a group must exist, and must have an associated access control level
2. Only the Administrator can CRUD group information

### 4.6.2 Actors

1. Administrator

### 4.6.3 Pre-Conditions

1. The Administrator is logged into the system.

### 4.6.4 Basic Path: Administrator Adds a Group

1. The Admin selects an option to add a new Group to the system
2. The System prompts the Admin for:
  - (a) The group name
  - (b) Access levels for all functional areas
    - i. These access levels default to an initial access level of "read-only"
3. The Admin enters the group name, and sets the desired access level for each functional area
4. The Admin saves this information
5. If the group name is invalid
  - (a) The system tells the Admin that the name is invalid, and they are given proper naming convention information
6. If the group name is already in use
  - (a) The system tells the Admin that the name is already in use
7. The system saves the new Group name, functional areas and access levels

## 4.6. Add Group Account

---

### Post-Conditions

1. A new Group is created on the system
2. The Group has a complete list of functional areas and their associated access levels
3. No users are assigned to the new group at this time
4. **The new group will show up in the "List Groups" use case**

### 4.6.5 "Used" Use Cases

1. List Groups

### 4.6.6 Other Requirements

None.

## 4.7 List Groups

This use case describes the process of listing groups in the ACME system.

### 4.7.1 Notes

1. This List Groups option can take the Administrator to the Edit Group use case

### 4.7.2 Actors

1. Administrator
2. External User

### 4.7.3 Pre-Conditions

1. The user is logged into the system.

### 4.7.4 Basic Path: User Displays a List of Groups

1. The user selects an option to list the groups on the system
2. For all users the system displays the complete list of group names
  - (a) Note - all group names are displayed without restriction
  - (b) Note - functional areas and their access levels for each group are not initially displayed here
3. The user can optionally select any group, and then display the functional areas and access levels for that group
4. If the user is the Administrator
  - (a) These functional areas and access levels are displayed in read/write mode
5. Else
  - (a) These functional areas and access levels are displayed in read-only mode

### Post-Conditions

1. The Administrator may be in a position to edit the functional areas and access levels of a group
2. Note - users cannot be deleted from groups here. That must occur at the "Edit User" screen.

#### 4.7. List Groups

---

##### 4.7.5 "Used" Use Cases

None.

##### 4.7.6 Other Requirements

None.

## 4.8 Edit a Group

This use case describes the process of editing a group in the ACME system.

### 4.8.1 Notes

### 4.8.2 Actors

1. Administrator
2. User

### 4.8.3 Pre-Conditions

1. The user is logged into the system.

### 4.8.4 Basic Path: Administrator Edits a Group

1. The Admin displays a list of groups
2. The Admin selects one group from the list, then selects an option to edit that group
3. The system displays the following required group information in read/write mode:
  - (a) The group name
  - (b) A list of the group functional areas and their corresponding access levels
4. The Admin can change the group name and edit the access level of the functional areas
5. The Admin saves the changes
6. If the group name is invalid
  - (a) The system tells the Admin the group name is invalid and displays proper format options
  - (b) When the Admin disposes the message
    - i. The system leaves the contents of the group name data entry field as-is
    - ii. The system places focus in the group name field
7. Else, the system saves the changes

### Post-Conditions

1. The group name may be different

## 4.8. Edit a Group

---

- (a) Any users in that group stay in that group, even though the group has a different name
2. Functional areas for the group may have different access levels
3. This will not affect users currently logged into the system
4. This will affect users the next time they log into the system

### 4.8.5 Alternate Path #1: Non-Administrator Views Group Information

1. The User displays a list of groups
2. The User selects one group from the list
3. The system makes a View option available to the user
4. When the user selects the View option the system displays the following required group information in read-only mode:
  - (a) The group name
  - (b) A list of the group functional areas and their corresponding access levels

#### Post-Conditions

1. There should be no data changes to any group
2. From an information standpoint, the user who viewed this information will know the access levels of any groups that they viewed

### 4.8.6 "Used" Use Cases

None.

### 4.8.7 Other Requirements

None.

## 4.9 Delete a Group

This use case describes the process of deleting a group in the ACME system.

### 4.9.1 Notes

1. None.

### 4.9.2 Actors

1. Administrator

### 4.9.3 Pre-Conditions

1. The Administrator is logged into the system.

### 4.9.4 Basic Path: Administrator Deletes One Group

1. The Admin selects an option to list all groups
2. The Admin selects a group, then selects an option to delete that group from the system
3. **If the Group contains any user accounts**
  - (a) The system prompts the Admin "The group contains user accounts and cannot be deleted"
  - (b) **The system lists the user account names**
  - (c) The system does not remove the group
4. **Else, if the group contains no user accounts**
  - (a) The system prompts the Admin "Are you sure?"
  - (b) The Admin replies "Yes"
  - (c) The system removes the group account from the database

### 4.9.5 Post-Conditions

1. **If the Group contained no user accounts it is deleted**

### 4.9.6 "Used" Use Cases

None.

## 4.9. Delete a Group

---

### 4.9.7 Other Requirements

1. The Administrator can delete groups that have no user accounts



## 4.10 Enable/Disable Security

This use case describes the process of enabling and disabling the security (access control) system.

### 4.10.1 Notes

None.

### 4.10.2 Actors

1. Administrator

### 4.10.3 Pre-Conditions

1. The Administrator is logged into the system.

### 4.10.4 Basic Path: Enabling Security

1. The use case starts with the system security in a "disabled" state
2. The Administrator selects an option to enable system security
3. The System sets an internal trigger to enable security for subsequent user logins

### Post-Conditions

1. When users next log in:
  - (a) They will have to enter a username and password to enter the system
  - (b) Their group permissions will be in effect
2. Users currently logged in are not affected

### 4.10.5 Basic Path: Disabling Security

1. The use case starts with the system security in a "enabled" state
2. The Administrator selects an option to disable system security
3. The System sets an internal trigger to disable security for subsequent user logins

#### 4.10. Enable/Disable Security

---

##### **Post-Conditions**

1. When users next log in:
  - (a) They *will not* have to enter a username and password to enter the system
  - (b) Their group permissions *will not* be in effect
2. Users currently logged in are not affected

##### **4.10.6 "Used" Use Cases**

None.

##### **4.10.7 Other Requirements**

None.

## 4.11 Become Administrator

This use case describes the process where any user – besides the Administrator – can attempt to become the Administrator. The motivation for this is:

1. It is preferred by ACME that the Login screen not be displayed when security is disabled
2. This leads to a condition where a user cannot login as the Administrator (because there is no Login screen) when security is disabled
3. Therefore, this use case is a workaround for not having a login screen but also needing to have an Administrator log in

### 4.11.1 Notes

1. This use case is the same whether security is enabled or disabled
2. **ISSUE:** This use case affects only the Editor (frame, or session) currently under consideration; if the user has multiple Editors (sessions) open at the same time, those other Editors are not affected by this change
3. **Administrator login attempts are not logged**

### 4.11.2 Actors

1. External user

### 4.11.3 Pre-Conditions

1. The user is logged into the system
2. Security may be enabled or disabled
  - (a) The difference here is that the original user id is either a real user like "Ralph" (if security is enabled) or the user will be the user "Nobody" (if security is disabled)

### 4.11.4 Basic Path: Become Administrator

1. The user selects an option to "Become Administrator"
2. If the system detects that an Administrator is already logged in:
  - (a) The system notifies the user of this condition (i.e., "The Administrator is already logged in, and can only log in once.")

#### 4.11. Become Administrator

---

- (b) The use case stops here
- 3. The system prompts the user for the Administrator password
- 4. If the password is invalid, the user is informed of this
  - (a) (There are no limits on the number of login attempts)
- 5. If the password is correct, continue
- 6. If the user does not have a current job open, continue
- 7. If the user has a current job open, and the job is clean:
  - (a) The system closes the job; continue
- 8. Else, if the user has a current job open, and the job is dirty:
  - (a) The system prompts the user to save their information ("Save your changes - yes, no, cancel?")
    - i. If the user selects Cancel, this operation is stopped
    - ii. If the user selects No, their changes are discarded; continue
    - iii. If the user selects Yes, their changes are saved; continue
  - (b) The system closes the job
- 9. The system changes the internal state of the program so that it is just as though the Administrator has just logged in, except:
  - (a) The system remembers who the user was before they became Administrator
- 10. If the username is displayed on screen, such as in a status bar, the username is changed to "Administrator"
- 11. **The user performs their administrative functions ...**
- 12. **The user attempts to switch back to their previous identity**
- 13. **They are returned to their previous identity**

#### Post-Conditions

- 1. The user has all the permissions of the Administrator
- 2. The system marks any database actions as being performed by the Administrator
- 3. **The user can "switch back" to their original user id**

#### 4.11.5 "Used" Use Cases

- 1. See the "External User Login" use case for the Administrator for the Administrator's initial login conditions regarding the Navigator and menu items that are enabled

#### 4.11.6 Other Requirements

1. A user cannot become the Administrator and then become another user
  - (a) An Administrator cannot become any other user at any time; only when this sequence of steps is followed can they switch back to being their original identity
  - (b) Note - that is not the intent of this use case
  - (c) The "Switch User" option is only available to non-Administrator users
  - (d) The "Switch Back" option is only available to the Administrator when they used the "Become Administrator" option to become the Administrator
2. If the user originally logged in as Administrator, the "Switch to Administrator" and "Switch Back" options are not available

## 4.12 Creating a New ACME Job

This use case describes the process of an external user creating a new ACME job.

### 4.12.1 Notes

1. TMF and Contract information is not needed for this phase (2c) for Unplanned jobs

### 4.12.2 Actors

1. External User
2. Administrator

### 4.12.3 Priority

High.

### 4.12.4 Status

High level.

### 4.12.5 Pre-Conditions

1. The ACME Editor is actively running when the use case begins
2. There may only be one ACME Editor open, or there may be multiple Editors open.
3. The user may have a job already open in the current Editor, or not.

### 4.12.6 Basic Path: No job is currently open in the ACME Editor

1. The user selects an option to create a new job
2. The system uses the "Prompt User for Job Information" sub use case
3. The user enters this information, then selects the Continue option
4. The system creates an entry for this job in the database
  - (a) If security is enabled, the user id is associated with the job
  - (b) If security is not enabled, *the user id of the user "Unknown User"* is associated with the job

#### 4.12. Creating a New ACME Job

---

5. The system puts the short description of the job in the title bar of the Editor
6. The system displays the Navigator with Navigation buttons activated

#### **Post-Conditions**

1. The new job is opened in the current ACME Editor
2. A new job entry is entered into the database
3. The "date created" and "date modified" for the job are marked with the same timestamp

#### **4.12.7 Alternative Path #1, Already a "clean" job open in the ACME Editor**

1. The user is in the ACME Editor, and has a clean job open in the Editor
2. The user selects an option to create a new job
3. The system uses the "Prompt User for Job Information" sub use case
4. The user enters this information, then selects the Continue option
5. The system prompts the user - "Open job in current Editor? Open a new Editor frame? Cancel?"
6. If the user selects Cancel
  - (a) The New Job operation is stopped
  - (b) The system displays the current ACME Editor in the state just prior to the start of the Job Open scenario
7. Else, If the user selects "Open job in current Editor"
  - (a) The old (clean) job is closed
  - (b) The system creates an entry for this job in the database
  - (c) The new job is opened in the current Editor
8. Else, if the user selects "Open job in a new Editor frame"
  - (a) The system creates a new Editor frame
  - (b) The system creates an entry for this job in the database
  - (c) Opens the job in that frame
  - (d) That frame is displayed in the foreground (in front of the previous ACME Editor frame)
9. The system puts the short description of the job in the title bar of the Editor

#### 4.12. Creating a New ACME Job

---

10. The system displays the Navigator with Navigation buttons activated

##### **Post-Conditions**

1. A new job entry is entered into the database
2. A new job is opened in either the current ACME Editor or in a new ACME Editor frame
3. The "date created" and "date modified" for the job are marked with the same timestamp

#### **4.12.8 Alternative Path #2, Already a "dirty" job open in the ACME Editor**

1. The user is in the ACME Editor, and has a "dirty" job open in the Editor
2. The user selects an option to create a new job
3. The system uses the "Prompt User for Job Information" sub use case
4. The user enters this information, then selects the Continue option
5. The system prompts the user - "Open job in current Editor? Open a new Editor frame? Cancel?"
6. If the user selects Cancel
  - (a) The New Job operation is stopped
  - (b) The system displays the current ACME Editor in the state just prior to the start of the "New Job" scenario
7. Else, If the user selects "Open job in current Editor"
  - (a) The user is informed "The current job has unsaved changes. Save changes, Discard changes, Cancel?"
  - (b) If the user selects "Cancel"
    - i. The process of creating the new job is canceled
    - ii. The user is returned to their Editor frame
  - (c) Else, if the user selects "Save"
    - i. The system saves the current job to the database
    - ii. The new job is opened in the current Editor frame
  - (d) Else, if the user selects "Discard"
    - i. The system discards the changes to the existing job
    - ii. The new job is opened in the current Editor frame
8. Else, if the user selects "Open job in a new Editor frame"



#### 4.12. Creating a New ACME Job

---

- (a) The system creates a new Editor frame
  - (b) The system creates an entry for this job in the database
  - (c) The system opens the job in the newly-created Editor frame
  - (d) That frame is displayed in the foreground (in front of the previous ACME Editor frame)
9. The system puts the name of the job in the title bar of the Editor
  10. The system displays the Navigator with Navigation buttons activated

#### **Post-Conditions**

1. A new job entry is entered into the database
2. A new job is opened in either the current ACME Editor or in a new ACME Editor frame
3. If a new frame was created for the new job, the previous job remains open in the previous Editor frame
4. The "date created" and "date modified" for the job are marked with the same timestamp

#### **4.12.9 Sub Use Case - Prompt User for Job Information**

1. The system prompts the user for the following required fields:
  - (a) Short Job Description
  - (b) Customer name
  - (c) Job trim size
  - (d) Magazine Type (S, T, or D)
2. The system prompts the user for the following optional fields:
  - (a) Long job description
  - (b) Title
  - (c) Issue
  - (d) Starting Folio No.

#### **4.12.10 "Used" Use Cases**

None.

#### **4.12.11 Other Requirements**

1. Any time the phrase "the system creates an entry for this job in the database" is seen in the previous statements in this use case:

#### 4.12. Creating a New ACME Job

---

- (a) If security is enabled, the user id is associated with the job
  - (b) If security is not enabled, the user id of the user "nobody" is associated with the job
2. The job type is always "Unplanned"
3. Multiple customer numbers may be present
  - (a) When there is only one customer show a *read-only textfield*
  - (b) When there are multiple customers display the customer information in a drop-down menu

## 4.13 Search/List ACME Jobs

1. This use case lets the user search for jobs that are currently on their system.
2. This use case does not really stand on its own, but is used by other use cases.

### 4.13.1 Notes

1. The order of the fields on the search screen, and their order in the search results table, is correct on the prototype, and may not be correct below
2. Note that this is technically a "sub use case", meaning that it is used by other use cases, such as "Open a Job", and "Delete a Job"
3. Therefore, per our discussion, I've grayed-out some lines below which technically should not be in this use case

### 4.13.2 Actors

1. External User
2. Administrator

### 4.13.3 Pre-Conditions

1. The ACME Editor is actively running when the use case begins

### 4.13.4 Non-Administrator Selects the Search/List Option

1. The user selects a "List Jobs" option
2. The system displays a window that lets the user filter job listings by:
  - (a) Customer
  - (b) Date modified (a range)
  - (c) Job short description
  - (d) Long description
  - (e) Title
  - (f) Created By
    - i. This is a drop-down list, and includes all users on the system, as well as an additional "All Users" selection
  - (g) Issue
  - (h) Type of magazine

#### 4.13. Search/List ACME Jobs

---

3. The user enters their desired search criteria, then selects a search option
4. The system lists all jobs without restriction, in sorted order by the following fields:
  - (a) *User that created the job*
    - i. *If the user account has been deleted from the system, display Unknown User instead*
  - (b) Customer
  - (c) Short description
  - (d) Title
  - (e) Issue
  - (f) Date modified
5. The user can sort the data by any of the displayed columns
6. The user can use standard "VCR buttons" to move to the First, Previous, Next, Last and User-specified page numbers
7. If the user is the Administrator
  - (a) Job CRUD options (Open, Delete) are disabled
  - (b) A "remove job locks" option is enabled
8. Else:
  - (a) A "remove job locks" option is enabled
    - i. Note: On a subsequent screen, users will only be able to remove their own job locks. They specifically cannot remove locks of other users.
  - (b) If security is disabled
    - i. All job CRUD options are available to the user
  - (c) If security is enabled
    - i. Job CRUD options will be made available in accordance with the user's job rights

#### Post-Conditions

1. The user is viewing a list of jobs on their system

#### 4.13.5 "Used" Use Cases

None.

#### 4.13. Search/List ACME Jobs

---

##### **4.13.6 Other Requirements**

None.

## 4.14 Open a ACME Job

This use case describes the process of an external user opening a ACME job.  
*This use case needs to be refactored and made more readable.*

Specific problems:

1. Users can have multiple ACME sessions open per PC
2. Users can be logged into more than one PC

### 4.14.1 Notes

1. This use case is complicated and may need to be refactored to be more readable
2. A Job module (such as Job Information/Characteristics) may be locked by another user
3. A Job may be opened by another user, but no modules may be currently locked by that user

### 4.14.2 Actors

1. External User
2. Administrator

### 4.14.3 Pre-Conditions

1. The ACME Editor is actively running when the use case begins
2. There may only be one ACME Editor open, or there may be multiple Editors open.
3. The user may have a job already open in the current Editor, or not.
4. In a networked environment, another user may have the job open
5. Security may be enabled or disabled

### 4.14.4 Basic Path - No job is open in the ACME Editor

1. The user is in the ACME Editor, but does not have a job open in the Editor
2. The user selects an option to open a job
3. The system invokes the List Jobs use case to let the user select a job to open
4. The user selects a job to open

- (a) ***If the Job Characteristics module in the job is locked by another user:***
    - i. The system warns the user that the module is locked by another user (and provides that user's name)
    - ii. The system opens the job in the current editor in read-only mode
  - (b) Else, if another user is in the job but has no modules locked:
    - i. The system warns the user which other users are editing this job at this time
    - ii. The system opens the job in the current editor
  - (c) Else (if the job is not currently opened by another user)
    - i. The system opens the job in the current editor
    - ii. The system records a note that this user opened the job at this date/time
    - iii. The system records a note that the user is currently editing the job
  - (d) In any case
    - i. The ACME Editor is displayed
    - ii. The ACME Navigator is displayed with it's options enabled
5. ***When the user selects the Edit Job (i.e., "job characteristics") option***
- (a) If the user has write permission in this functional area
    - i. If the module is not locked by another user
      - A. The system locks the job at the Edit Job level
      - B. The user modifies the job information as desired
      - C. When the user selects an option to save and close the Job Characteristics, the Job Characteristics module lock is released
  - (b) Else, if the user has read-only permission in this area
    - i. The system opens to the Edit Job information in read-only mode
  - (c) ***Else, if the user has "hidden" access to job information, they are not allowed to open the Job Characteristics screen***
    - i. Either the Job Information button is deactivated, or the permission takes effect when the user clicks the button

**Post-Conditions**

1. A new job is opened in the current ACME Editor
2. The system is aware that this user is currently editing this job
3. If the user has Edit permission on the Job Information, and the user is in the Edit Job Information module, the system will maintain information to know that this user has this module locked

**4.14.5 Alternative Path #1, There is already a "clean" job open in the ACME Editor**

1. The user has a clean job open in an Editor, and the selects an option to open a job
2. The system invokes the List Jobs use case to let the user select a job to open
3. The user selects an option to open the selected job
4. The system detects that the job is currently being edited by another user
5. The system prompts that "Note: The following users also have the job open: [list users]. Continue opening, or Cancel?"
  - (a) If the user select Cancel, the job-open process is stopped at this point
  - (b) Else, if the user selects "Continue opening"
    - i. Continue with the rest of this use case (but with the job in "read-only" mode)
6. The system prompts the user - "Open job in current Editor? Open a new Editor frame? Cancel?"
  - (a) If the user selects Cancel, the Job Open operation is stopped, and system displays the current ACME Editor in the state just prior to the start of the Job Open scenario
  - (b) Else, If the user selects "Open job in current Editor"
    - i. The old/current (clean) job in the Editor is closed
    - ii. The new job is opened in the current Editor
  - (c) Else, if the user selects "Open job in a new Editor frame"
    - i. The system invokes the "Open Job in New Editor" sub use case
  - (d) The system records a note that this user opened the job at this date/time



#### 4.14. Open a ACME Job

---

7. If the user has Edit permission on the Job Information, and the user is in the Edit Job Information module, the system will maintain information to know that this user has this module locked

#### Post-Conditions

1. A new job is opened in the current ACME Editor or in a new ACME Editor frame

#### 4.14.6 Alternative Path #2, Already a "dirty" job open in the ACME Editor

*TODO - AL - MAY NEED TO RE-WORD THIS BELOW HERE*

1. The user is in the ACME Editor, and has a dirty job open in the Editor
2. The user selects an option to open a job
3. The system invokes the List Jobs use case to let the user select a job to open
4. If the job is currently locked by another user
  - (a) The system prompts "The job is open by another user; Open Read-Only, or Cancel?"
  - (b) If the user select Cancel, the job-open process is stopped at this point
  - (c) Else, if the user selects "Open Read Only"
    - i. Continue with the rest of this use case (but with the job in "read-only" mode)
5. If the job is not currently locked by another user
  - (a) The system prompts the user - "Open job in current Editor? Open a new Editor frame? Cancel?"
  - (b) If the user selects "Cancel"
    - i. The Job Open operation is stopped
  - (c) Else, If the user selects "Open job in current Editor"
    - i. The system tells the user that "The current job *has been modified*; save, discard, or cancel?"
    - ii. If the user selects "Cancel"
      - A. The Open Job operation is cancelled
      - B. The user is returned to their previous job
    - iii. If the user selects "Discard current job"

#### 4.14. Open a ACME Job

---

- A. The system discards the changes to the current job
- B. The system opens the new desired job in the current Editor
- iv. If the user selects "Save"
  - A. The current job is saved
  - B. The system opens the new desired job in the current frame
  - C. *The system creates an informational lock at the job level indicating that the job is now opened by this user*
  - D. The user begins working with the job ...
- (d) Else, if the user selects "Open job in a new Editor frame"
  - i. The system invokes the "Open Job in New Editor" sub use case
- (e) The system records a note that this user opened the job at this date/time
- (f) The system locks the job so that no other user can edit it at this time

#### Post-Conditions

1. The job open in ACME before the "Job Open" use case may or may not be saved to the database
2. A job is retrieved from the database and is opened in the current ACME Editor or in a new ACME Editor
3. A job-level informational lock is obtained on the newly opened job

#### 4.14.7 Sub Use Case: Open Job in New Editor

1. The system creates a new Editor frame
2. The system opens the desired job in that new frame
3. The system displays the new frame in the foreground, in front of the previous ACME Editor frame

#### 4.14.8 "Used" Use Cases

1. Search/List ACME Jobs

#### 4.14.9 Other Requirements

1. *When a user selects a job to open, if that job is already open in a different ACME Editor frame, bring that job to the foreground*
2. The system shall record a "last maintained by" user id field at the *job* level any time a user saves information in any module
  - (a) This will include the user id and and timestamp
  - (b) If security is not enabled, the user id of the Unknown User will be stored in the "last maintained by" field
3. The system shall record a "last maintained by" user id field at the *module* level any time a user saves information in a module
  - (a) This will include the user id and and timestamp
  - (b) If security is not enabled, the user id of the Unknown User will be stored in the "last maintained by" field
4. **Recent file list**
  - (a) The system will display a list of the user's most recently opened 10 jobs
  - (b) A job will be added to the list any time it is opened into the ACME Editor
  - (c) This list will appear under the Job menu, in a manner similar to the file history in Microsoft Word
  - (d) In the Job menu, the job list will only show the job short description
  - (e) This list will be stored on the user's computer, in a data file under the user's home directory
  - (f) Jobs will appear once and only once in the list
    - i. i.e., if the job has been opened three consecutive times, it should only appear once in the list
  - (g) The job list cannot be modified by the user
  - (h) Note: This job list may get out of sync with jobs that are actually in the database, and this must be accounted for
  - (i) This requirement is not documented in the scenarios above
    - i. The behavior when a job in this list is accessed should be the same as the "Open Job Scenario"

New section  
and line  
items

#### 4.14. Open a ACME Job

---

##### 4.14.10 Issues

1. If the user has a job open on one PC, then moves to another PC, they may have the Job Information module locked from the first PC. How is this handled on the second PC?
  - (a) Per our discussion of 2004-01-27, this will be handled in "the most simple" manner (TBD)

## 4.15 Save a ACME Job

This use case documents two sequences of events related to the saving a job for an external user, specifically:

1. Job — Save

### 4.15.1 Notes

1. *The Administrator has been removed as an actor for this use case*

### 4.15.2 Actors

1. External User

### 4.15.3 Pre-Conditions

1. The ACME Editor is open

### 4.15.4 Basic Path: User Saves Job

1. The user has a job open in read-write mode, and it is in a "dirty" state (i.e., the Job Characteristics have been changed)
  - (a) [An implication here is that the user has Edit access in the "Save Job" functional area](#)
2. When the system enters a dirty state, the system enables the "Save Job" option
3. The user selects the "Save Job" option
4. The system saves the job details for this job
5. The system saves the "last maintained by" information:
  - (a) [At the job level](#)
  - (b) [At the module level](#)
6. The system changes the Editor state to "clean"
7. The system disables the "Save Job" option
8. If the user exits the Job Information module, the Job Information lock for this job is cleared
9. Else, the Job Information lock for this job *is not* cleared

**Post-Conditions**

1. The job is saved to the database
2. The Editor state is changed to "clean"
3. The lock for the Job Information module is released if the user disposes that screen

**4.15.5 Alternative Path #1, The Save As ... option**

1. If the user has "Job - Edit", the system enables the "Save Job As ..." option
2. The user selects the "Save Job As ..." option
3. The system prompts the user for a short description for the job
  - (a) Note - no other job-related criteria can be modified here
4. If the short description is not unique:
  - (a) **Note: Whether or not the job was created by the current user, continue (many changes here)**
  - (b) If the user does not have "delete job" authority
    - i. The system prompts the user "The short description you have supplied is not unique, and you do not have permission to overwrite a job." The use case is stopped.
  - (c) Else, if the user has "delete job" authority, continue
  - (d) The system displays a message: "A job already exists with this name; overwrite, cancel?"
  - (e) If the user selects cancel, they are returned to the previous screen where they enter the short description
  - (f) Else, if the user selects "overwrite"
    - i. The system deletes the old job details, and creates a new job with the current job information
5. Else, if the short description is unique, the system saves the job with that name
6. The job in the Editor is now the job with the new short description
  - (a) The new short description is displayed in ACME's title bar
  - (b) The originating job is closed, and none of the changes are saved to that job

#### 4.15. Save a ACME Job

---

##### Post-Conditions

1. The original job data is saved to the database under a new short description
2. The "new" job is now in the ACME Editor
3. *Note: the process of prompting the user to save changes to the originating job is specifically not desired*

##### 4.15.6 "Used" Use Cases

##### 4.15.7 Other Requirements

1. The Save option is only activated when a job is dirty

##### 4.15.8 Issues

###### *New*

1. *"Save As" and Overwrite imply the ability to delete previous jobs, hence the need for delete permission*
2. *Implications:*
  - (a) *In single-user mode*, the Unknown User can delete Unknown User jobs
  - (b) *In multi-user mode*, assuming that a user has Edit access to the "Job - Delete" functional area:
    - i. Margaret can delete Ralph jobs, etc.

## 4.16 Delete a Job

In this use case a user deletes one or more jobs.

### 4.16.1 Notes

1. *Administrator has been removed as an actor, and cannot participate in this use case*

### 4.16.2 Actors

1. External user

### 4.16.3 Pre-Conditions

1. ACME is running on the user's system.

### 4.16.4 Basic Path - User deletes one job

1. The user uses the "List Jobs" use case
2. The user selects one job they want to delete
3. If security is enabled
  - (a) If the user *does not* have "delete" permission
    - i. The system does not enable the Delete option for this job; stop here
  - (b) *Else, if the user does have delete permission, continue*
4. Else, (if security is not enabled) continue
5. The system makes the Delete option available to the user
6. The user selects the Delete option
7. If the system detects that a user has a lock on a module for this job:
  - (a) The system displays a message indicating "The job cannot be deleted at this time *because the MODULE\_NAME module* is locked by 'USERNAME'"
  - (b) The use case is stopped here
8. Else, if the system detects that a user has an information lock on the job:
  - (a) The system displays a message "The user 'USERNAME' currently has this job open, and it cannot be deleted."
  - (b) The use case is stopped here



#### 4.16. Delete a Job

---

9. Else, continue
10. The system prompts "You are about to delete 'JOBNAME' – are you sure? (y/n)"
11. If the user selects "No"
  - (a) The system cancels the delete attempt
  - (b) The job in the list remains highlighted
12. Else, if the user selects "Yes, I am sure", continue
13. Else, the system removes the job and all job details from the database
14. The system refreshes the list of jobs

#### Post-Conditions

1. The selected job is removed from the ACME system
2. The "job list" screen is updated if it is visible

#### 4.16.5 Alternative Path #1, User selects multiple jobs to delete

1. The user uses the "List Jobs" use case
2. The user selects two or more jobs that they want to delete
3. If security is enabled
  - (a) If the user does not have "delete" permission
    - i. The system does not enable the Delete option
    - ii. (The use case stops here)
  - (b) Else, if the user has delete permission, continue
4. Else, if security is not enabled, continue
5. The user selects the "Delete Job" option
6. The system prompts "You are about to delete the following jobs" (followed by a list of job names); "Are you sure?"
7. If the user selects "No", the system cancels the delete attempt, and the jobs remain highlighted
8. Else, if the user selects "Yes, I am sure"
9. ***The system deletes all the jobs where no job or module locks exist***
10. ***If there are any jobs that have informational locks or module locks***
  - (a) The system displays a message indicating "The following jobs are locked and cannot be deleted at this time", followed by the short

#### 4.16. Delete a Job

---

description for each job, the module that is locked, and the user that has the lock

11. The system refreshes the list of jobs

#### Post-Conditions

1. One or more jobs are removed from the ACME system
2. The "job list" screen is updated if it is visible
3. Jobs with informational locks or module locks are not deleted

#### 4.16.6 "Used" Use Cases

1. Search/List Jobs

#### 4.16.7 Other Requirements

1. *With security disabled all users have "Delete Job" permission*
2. *With security enabled, certain users will have "Delete Job" permission. Those users can delete jobs:*
  - (a) Created by themselves
  - (b) Created by other users such as Margaret, Ralph, etc.
  - (c) Created by Unknown User

## **4.17 Clone a ACME Job**

This option is no longer available, but the same functionality is available through the "Job Save As ..." use case.

## 4.18 Clear Job Lock

This use case describes the process of clearing a lock on a job. This can potentially occur when a job is open by a user, and then something happens to that user's computer, such as the computer losing power.

### 4.18.1 Notes

1. *The Administrator can clear a lock on any job*
2. In single-user mode, a user can clear a lock on any job
3. In multi-user mode, other users can clear locks on their jobs
4. In multi-user mode, other users with Edit access to the "Clear Job Locks" functional area can clear locks on other user's jobs

### 4.18.2 Actors

1. Administrator
2. External User

### 4.18.3 Pre-Conditions

1. The user is logged into the system
2. A user has come to the administrator telling them that they cannot edit a job because there is a false lock on the job

### 4.18.4 Basic Path - Administrator Clears a Job Lock

*Many changes in this area*

1. The Administrator searches for the job, then selects a "Clear Job Locks" option
2. The system displays a list of locks on this job, including:
  - (a) The username
  - (b) Job level informational locks
  - (c) Locks on job modules
  - (d) Timestamps for each lock
3. The administrator selects the locks they want to clear
4. After an "Are You Sure?" prompt, the system clears the locks

#### 4.18.5 Alternate Path #1 - Non-Administrator Clears a Job Lock

*All new*

1. A user searches for a job, then selects a "Clear Job Locks" option
2. The system displays a list of locks on this job, including:
  - (a) The username
  - (b) Job level informational locks
  - (c) Locks on job modules
  - (d) Timestamps for each lock
3. If the locks on the job are not by this user, and the user does not have Edit permission for the "Clear Job Locks" functional area, they cannot select any of the locks. The use case stops here.
  - (a) Note the problem for the Unknown User user
4. Else, if there are one or more locks on this job, or the job is locked by the Unknown User user account they can select the desired locks, then select a Clear Locks option
5. After an "Are You Sure?" prompt, the system clears the locks

#### Post-Conditions

1. The locks on a job and job modules for the current user have been removed

#### 4.18.6 "Used" Use Cases

1. List/Search Jobs

#### 4.18.7 Other Requirements

#### 4.18.8 Issues

1. How to handle this in single-user mode?
  - (a) The Unknown User can unlock any job

## 4.19 Exiting the Application

This use case defines the sequence of events that occurs when the user exits a session (Editor).

### 4.19.1 Notes

1. The "Close All Editors" use case has been removed from this documentation. Each Editor must now be closed individually.

### 4.19.2 Actors

1. External User
2. Administrator

### 4.19.3 Pre-Conditions

1. The ACME system is running on the user's system
2. The user may have one or more ACME sessions running in separate ACME Editor frames

### 4.19.4 Basic Path: One Editor Frame is Open

1. The user is using ACME, with one Editor frame open
2. The user elects to "Close" the ACME Editor
3. If there is no job in the Editor
  - (a) The system shuts itself down
4. Else, if the job in the Editor is clean
  - (a) *The system closes the ACME Editor frame (without prompting the user)*
  - (b) *The system removes any locks showing that this user had this job open*
  - (c) The system shuts itself down
5. Else, if the job in the Editor is dirty
  - (a) The system prompts the user that "The job 'JOBNAME' has been modified, save changes - y/n/cancel?"
    - i. If the user selects "cancel", the shutdown operation will be stopped
    - ii. If the user selects "yes"

#### 4.19. Exiting the Application

---

- A. The dirty job is saved
  - B. Job module locks are released
  - C. This ACME Editor is closed
  - D. The system removes the database entry showing that this user had this job open
  - E. The system shuts itself down
- iii. If the user selects "no"
- A. The dirty job *is not* saved
  - B. Job module locks are released
  - C. This ACME Editor is closed
  - D. The system removes the database entry showing that this user had this job open
  - E. The system shuts itself down

#### **Post-Conditions**

1. Job module locks are released (if they had been acquired)
2. The informational lock on the job is released

#### **4.19.5 "Used" Use Cases**

N/A

#### **4.19.6 Other Requirements**

**These are all new**

1. The option to close a ACME Editor shall be named "Close", and will appear under the "Job" menu
2. Any time a job is closed:
  - (a) Locks on job modules shall be released
  - (b) The informational lock on the Job (proper) shall be released

## 4.20 Add a ACME Database

This use case describes the process of adding a ACME database to the list of known databases.

### 4.20.1 Notes

1. Any user can create a new database alias
2. Users do not share database aliases; they are saved on their local computers under their local accounts
3. Database definitions will be stored under the user's home directory on their computer
  - (a) Ramification - This database list is available only for this user on this PC

### 4.20.2 Actors

1. External User
2. Administrator

### 4.20.3 Pre-Conditions

1. ACME is installed

### 4.20.4 Basic Path

1. The user starts ACME, but initiates the action to bring up the database management screen
2. The system invokes the "List Database Aliases" option
3. The user selects an "Add Database" option
4. The system prompts the user for the following required fields:
  - (a) Database name/alias (must be unique)
  - (b) Server/system where the database is installed
  - (c) Database user id
  - (d) Database password
5. The user optionally tests the database connection
  - (a) The system informs the user of the success/failure of the connection
6. The user saves the new database connection



#### 4.20. Add a ACME Database

---

##### **Post-Conditions**

1. An alias for a new database connection is created and stored

##### **4.20.5 "Used" Use Cases**

1. Search/List Database Connections

##### **4.20.6 Other Requirements**

1. All fields are required
2. Database alias names are unique

## 4.21 Delete Database Alias

**This use case is new.**

This use case describes the process of deleting a database alias.

### 4.21.1 Notes

1. None

### 4.21.2 Actors

1. External User
2. Administrator

### 4.21.3 Pre-Conditions

1. ACME is installed

### 4.21.4 Basic Path

1. The user starts ACME, but initiates the action to bring up the database management screen
2. The system invokes the "List Database Aliases" use case
3. The user selects a database alias to delete, then selects the Delete option
4. The system prompts with an "Are You Sure?" message before allowing the deletion

### Post-Conditions

1. The database alias is removed

### 4.21.5 "Used" Use Cases

1. List Database Aliases

### 4.21.6 Other Requirements

None.

## 4.22 Edit a ACME Database Alias

**This use case is new.**

This use case describes the process of editing a ACME database alias.

### 4.22.1 Notes

### 4.22.2 Actors

1. External User
2. Administrator

### 4.22.3 Pre-Conditions

1. ACME is installed
2. One or more aliases have already been created

### 4.22.4 Basic Path

1. The user starts ACME, but initiates the action to bring up the database management screen
2. The system invokes the "List Database Aliases" option
3. The user selects an "Edit Database" option
4. The system displays the Edit Database screen
5. The user makes changes to the database alias and elects to save the changes
6. The system validates the changes, then updates the list to reflect the changes

### Post-Conditions

1. Any of the fields for the database alias may be modified

### 4.22.5 "Used" Use Cases

1. Search/List Database Connections

### 4.22.6 Other Requirements

1. All fields are required
2. Database alias names are unique

## 4.23 List Known Database Aliases

**This use case is new.**

This use case describes the process of listing the known database aliases for a given user.

### 4.23.1 Notes

1. None

### 4.23.2 Actors

1. External User
2. Administrator

### 4.23.3 Pre-Conditions

1. ACME is installed

### 4.23.4 Basic Path

1. The user starts ACME, but initiates the action to bring up the database management screen
2. The system displays a list of all known database aliases (without any filtering)
3. The system displays the following command actions to the user:
  - (a) Add
  - (b) Delete
  - (c) Edit

### Post-Conditions

1. A full list of database aliases is presented to the user

### 4.23.5 "Used" Use Cases

None.

### 4.23.6 Other Requirements

None.

#### 4.23. List Known Database Aliases

---

# Chapter 5

## Database Design

The database has not been designed in detail yet, but should include at least the following tables:

1. Customers
2. Jobs
3. Issues
  - (a) A subset that includes only the fields needed for Phase 2C
4. Application\_Data
  - (a) Single-user or multiuser mode
5. Users
  - (a) Username
  - (b) First name
  - (c) Last name
  - (d) Middle Initial
  - (e) Group ID
  - (f) Password
6. Groups
  - (a) Group ID
  - (b) Group Name
7. Functional-Areas
  - (a) FA ID
  - (b) FA Name
8. Groups-to-Functional-Areas

- 
- (a) Group ID
  - (b) FA ID
  - (c) Right/permission (hide/view/edit)

# Chapter 6

## User Interface

### 6.0.7 Menu Design

This section defines the pull-down menus that will appear in ACME. Note that all menu items may not be visible to the user. This will depend on who the user is, and what their security settings are.

1. Job
  - (a) New
  - (b) Open
  - (c) Delete
  - (d) Save
  - (e) Save-As
  - (f) Exit
2. Administration
  - (a) User Administration
  - (b) Settings
  - (c) Become Administrator
  - (d) Switch Back

### 6.0.8 Prototypes

The current UI prototypes are being delivered in a separate document.



# Chapter 7

## Issues

This section lists all known currently unresolved issues.

**This section is entirely new.**

1. Question: Can you have two users, such as Ralph/Active and Ralph/Inactive?
  - (a) **Answer: No (resolved 1-20-2004)**
2. Multiuser mode
  - (a) When multiple users use the system, the system should be in multiuser mode, and security needs to be enabled, and real usernames must be used
  - (b) This is not going to be significantly enforced until Phase 2E, and is a potential cause for error until that time
3. Data entry validation is not detailed yet
  - (a) Starting folio number is assumed to be like one of the following:
    - i. 1
    - ii. A
    - iii. 1A
    - iv. 3a
    - v. A2
    - vi. a4
4. The database is not detailed yet
  - (a) This will be resolved during the week starting 1-20-2004
  - (b) This will not affect the bid price, but is required to enable the ability to track changes
5. **Documentation format is TBD.**

- 
6. No ability to switch between single-user mode and multi-user mode is provided in Phase 2C.
  7. In the "Open a Job" use case, handling the conflict between identifying one user opening the same job on multiple PCs has not been fully resolved. The agreement at this time is to do this in the most simple/straightforward way possible.

## Chapter 8

# Risks

This section attempts to identify and quantify risks that may be associated with this effort.

1. Any change in key personnel.
2. We go too far in this phase of development, and develop functionality that may not be used later.
3. We need new hardware in a timely manner.

# Index

- assumptions, 9
  - database, 11
  - service, 12
  - testing, 12
- documentation, 11
- help system, 11
- modes of operation, 10