Infor Infinium Application Manager Extended Programmer's Technical Guide

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

Publication Information

Release: Infor Infinium Application Manager Extended 3.0

Publication date: July 24, 2015

Document code: INFAM4_ALL_01

Table of Contents

About This Guide

This section focuses on the following information:

- Purpose of this guide
- Conventions used in this guide
- Organization of this guide
- Conventions used in this guide
- Related documentation

Intended audience

This guide is for the Infinium Application Manager developers who are responsible for using Infinium AM as developers.

Purpose of this guide

You should use this guide as a reference at your site and also to complement the instructor's presentation during a portion of the Infinium AM technical training course.

Organization of this guide

This guide is task oriented. We have grouped related tasks into chapters. Each chapter contains overview information and step-by-step instructions to lead you through the tasks.

Conventions used in this guide

This section describes the following conventions we use in this guide:

- Fonts and wording
- Function keys
- Character-based and graphical-based screens
- Prompt and selection screens
- Promptable fields
- Infinium applications and abbreviations

Fonts and wording

Convention	Description	Example
Italic typeface	Menu options and field names The guide uses the same abbreviations as the screen.	*Work With Controls* Use Max Lnth to specify the maximum length of alpha user fields.
Bold standard typeface	Used for notes, cautions and warnings	**Caution:** You must ensure that all Infinium AM users are signed off before reorganizing and purging. If there are jobs in the queue, those files will not be reorganized.

Convention	Description	Example
Bold typeface	Characters that you type and messages that are displayed	Type **A** to indicate that the position is alphanumeric and type **N** to indicate that the position is numeric. The following message is displayed: **Company not found**
F2 through F24	Keyboard function keys used to perform a variety of commands.	Press F2 to display a list of available function keys.
F13 through F24	Function keys higher than F12 require you to hold down the Shift key and press the key that has the number you require minus 12.	Press F19 to work with project and activity comments.
Select	Choose a record or field value after prompting.	Select **C** (capitalization), **E** (expense) or **B** (both) as the *Capitalization code* value.
Press Enter	Provide information on a screen and when you have finished, press Enter to save your entries and continue.	Press Enter to save your changes and continue.
Exit	Exit a screen or function, usually to return to a prior selection list or menu. May require exiting multiple screens in sequence.	Press F3 to return to the main menu.
Cancel	Cancel the work at the current screen (page) or dialog box, usually to return to the prior screen (page).	Press F12 to cancel your entries.

Convention	Description	Example
Help	To access online help for the current context (menu option, screen or field), press Help (or the function key mapped for help). To move through the other applicable levels of help, press Enter at each help screen. To return directly to the screen from which you accessed help, exit the help screen by clicking Exit or by pressing F3.	Press Help for more information about the current field.
[Quick Access Code]	Quick access codes provide direct access to functions. Most quick access codes in Infinium AM consist of the first letter of each word of the menu option name.	Work with sets [WWS]
Publication and course titles	Unless otherwise stated, titles refer to Infinium applications and use standard name abbreviations.	*Infinium Application Manager Guide to Basics* is referred to as *Infinium AM Guide to Basics.*

Function keys

Infinium AM function keys and universal Infinium AM function keys for the System i are described in the table below. All Infinium AM function keys are identified at the bottom of each screen.

Function Key	Name	Description
F1	Help	Displays help text
F2	Function keys	Displays window of valid function keys
F3	Exit	Returns you to the main menu

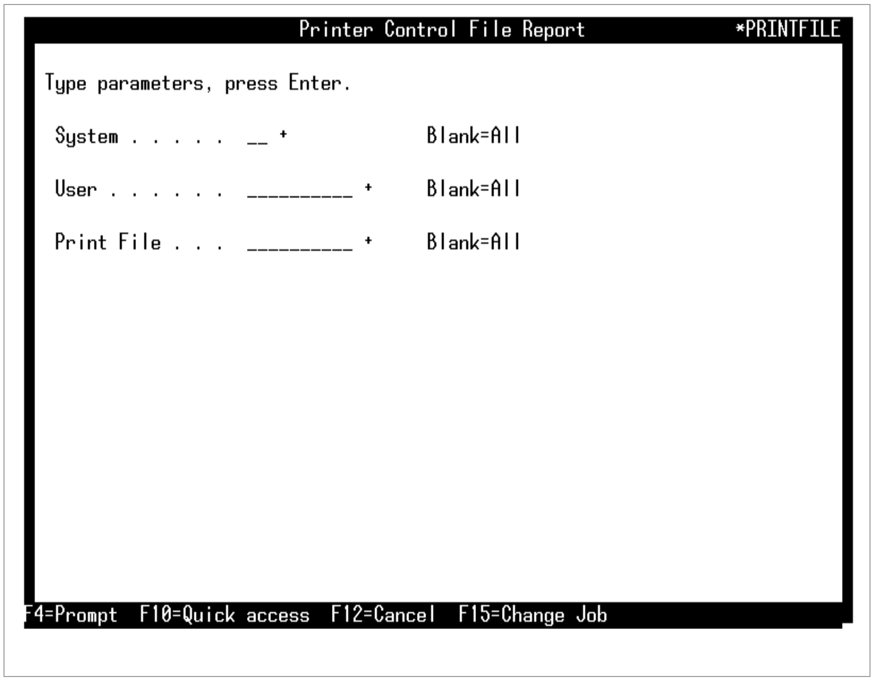
Function Key	Name	Description
F4	Prompt	Displays a list of values from which you can select a valid entry
F10	Quick Access	Enables you to access another function from any screen Type the quick access code in *Level*. You can change the application designator, such as PA, GL, IC and so forth, by selecting another application.
F12	Cancel	Returns you to the previous screen
F22	Delete	Deletes selected item(s)
F24	More keys	Displays additional function keys at the bottom of the screen

Prompt and selection screens

A prompt screen, similar to Figure 1, is the screen in which you type information to access a record or a subset of records in a file.

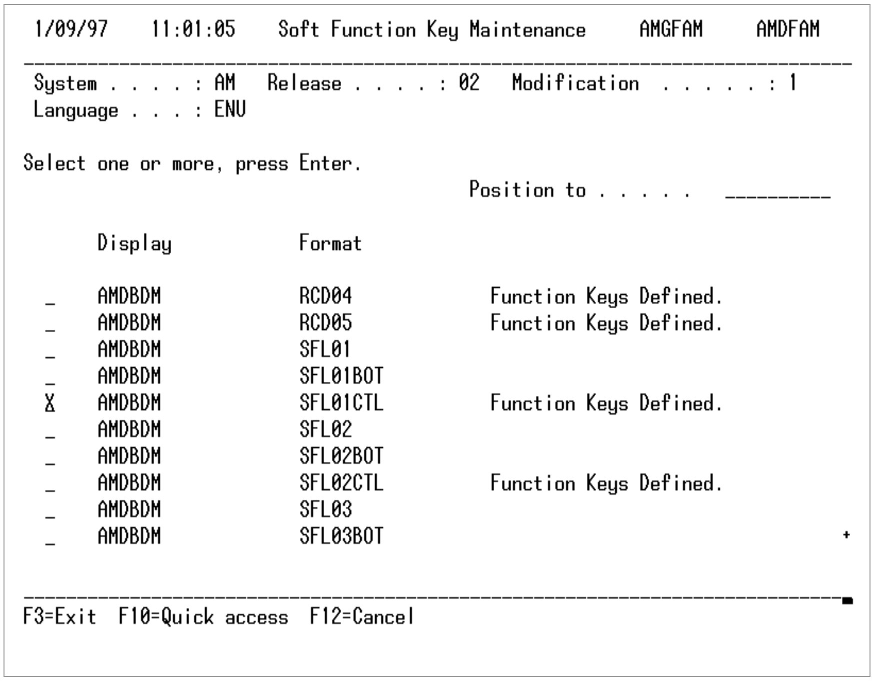
A selection screen, similar to Figure 2, is the screen from which you select a record or records to perform an action.

When we first explain a task in this guide, we fully document how you access a prompt and selection screen. If a related task uses that prompt or selection screen, we include the prompt and selection steps in that task. However, we do not include the screen again.



```
                         Printer Control File Report          *PRINTFILE

 Type parameters, press Enter.

  System . . . . .  __ +             Blank=All

  User . . . . . .  __________ +     Blank=All

  Print File . . .  __________ +     Blank=All















F4=Prompt  F10=Quick access  F12=Cancel  F15=Change Job
```

Figure 1: Prompt screen



```
 1/09/97    11:01:05    Soft Function Key Maintenance     AMGFAM     AMDFAM
______________________________________________________________________________
 System . . . . : AM   Release . . . . : 02   Modification  . . . . . : 1
 Language . . . : ENU

Select one or more, press Enter.
                                           Position to . . . . .   __________

       Display            Format

  _    AMDBDM             RCD04              Function Keys Defined.
  _    AMDBDM             RCD05              Function Keys Defined.
  _    AMDBDM             SFL01
  _    AMDBDM             SFL01BOT
  X    AMDBDM             SFL01CTL           Function Keys Defined.
  _    AMDBDM             SFL02
  _    AMDBDM             SFL02BOT
  _    AMDBDM             SFL02CTL           Function Keys Defined.
  _    AMDBDM             SFL03
  _    AMDBDM             SFL03BOT                                         +

______________________________________________________________________________
F3=Exit  F10=Quick access  F12=Cancel
```

Figure 2: Selection screen

Promptable fields

A plus sign displayed next to a field indicates that you can choose your entry from a list of possible values. Place the cursor in the field and press F4 to display a list of values.

To select an entry perform one of the following:

- Position the cursor at the desired value, type 1 and press Enter.
- Type the value in the appropriate field.

Infinium applications and abbreviations

The following table lists Infinium names and the corresponding product abbreviations that are associated with this product.

Application	Abbreviation
Infinium Application Manager Infinium Application Manager Extended	Infinium AM Infinium AM/X
Infinium Query Infinium Query Extended	Infinium QY Infinium QY/X
Infinium Financial Management Suite	**Infinium FM**
Infinium Accounts Receivable	Infinium AR
Infinium Cashbook	Infinium CB
Infinium Currency Management	Infinium CM
Infinium Financial Products	Infinium FP
Infinium Fixed Assets	Infinium FA
Infinium General Ledger	Infinium GL
Infinium Global Taxation	Infinium GT
Infinium Income Reporting	Infinium IR
Infinium Payables Ledger	Infinium PL
Infinium Project Accounting	Infinium PA
Infinium Purchasing/Payables Exchange	Infinium PX
Infinium ReportWriter	Infinium RW
Infinium Human Resources Suite	**Infinium HR**
Infinium Flexible Benefits	Infinium FB
Infinium Human Resources	Infinium HR

Application	Abbreviation
Infinium Human Resources/Payroll	Infinium HR/PY
Infinium Human Resources International	Infinium HR/UK
Infinium Payroll	Infinium PY
Infinium Training Administration	Infinium TR
Infor Human Capital Management Infinium Self-Service	Self-Service
Infinium Materials Management Suite	**Infinium MM**
Infinium Cross Applications	Infinium CA
Infinium Electronic Exchange	Infinium EX
Infinium Inventory Control	Infinium IC
Infinium Journal Processor	Infinium JP
Infinium Order Processing	Infinium OP
Infinium Purchase Management	Infinium PM
Infinium Process Manufacturing Suite	**Infinium PR**
Infinium Advanced Planning	Infinium MP
Infinium Formula Management	Infinium PF
Infinium Laboratory Management	Infinium LA
Infinium Manufacturing Control	Infinium MC
Infinium Regulatory Management	Infinium RM

Related documentation

For additional information about Infinium AM, refer to the following:

- *Guide to Infinium AM*
- *Infinium AM Guide to Basics*
- *Infinium AM Technical Guide*
- *Infinium AM Quick Reference Card*
- Online help

Installation instructions and release notes are available on Infor365.

Chapter 1 Overview of Infinium AM

1

This chapter contains an overview to the Infinium AM developer's information. In this chapter you learn about technical concepts that are pertinent to understanding Infinium AM from a developer's perspective.

The chapter consists of the following topics:

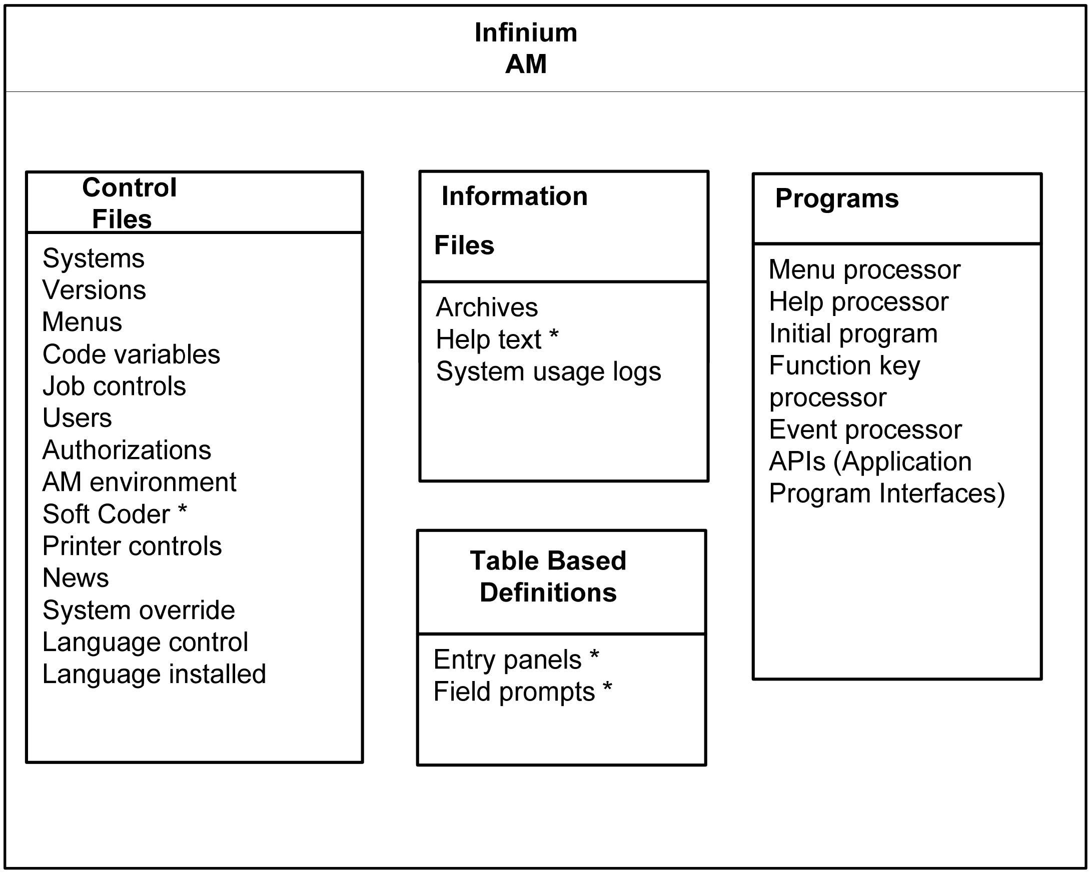
System overview

Infinium AM overview

As depicted in the diagram below, Infinium AM is a library that consists of programs, files, and tables that:

- Control processing functions common to all applications
- Help standardize the interface of applications to end users
- Minimize programming for functions that are used across applications, such as function keys and prompts
- Modularize and standardize the approach to programming new applications

Many of the Infinium AM features can be handled through options on the Infinium AM menu. However, additional programming is required to implement them.



* Areas marked with an asterisk require programming

Figure 1-1: Infinium AM overview

Implementing Infinium AM

Infinium recommends using the following order for implementing Infinium AM. You may choose to use a different order to fit your needs.

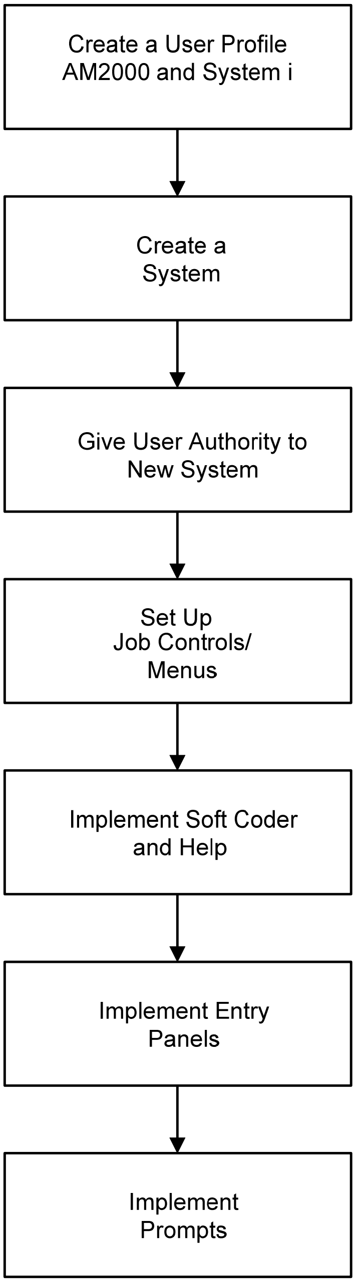


Figure 1-2: Implementing Infinium AM

Terminology and concepts

This section contains the Infinium AM terminology you should understand before you continue to the remaining chapters.

Systems and versions

The following is true of a system:

- Equivalent to an application
- Highest level in the Infinium AM hierarchy
- Identified with a two-character system designator

The following is true of a version:

- Accommodates multiple releases of an application
- Accommodates different database and program libraries
- Accommodates different menu structures

Release and modifications

The following is true of a release/mod*:*

- The application release number
- Help text, job controls, and function keys are release/mod sensitive

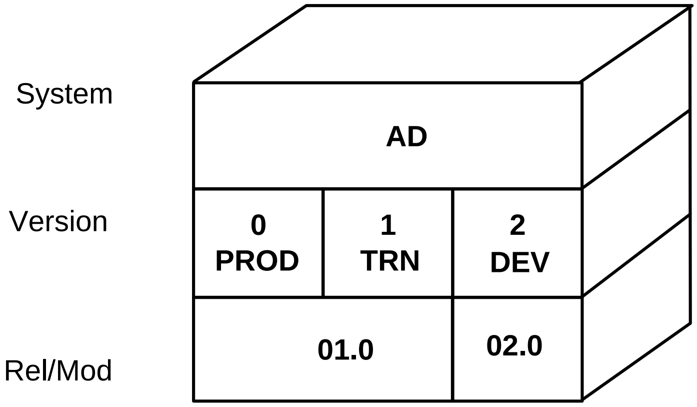


Figure 1-3: Systems, versions, release and modification

Menus and library lists

The following diagram depicts the hierarchy of menus and library lists. Each additional level of menus or code variables you create overrides the previous level.

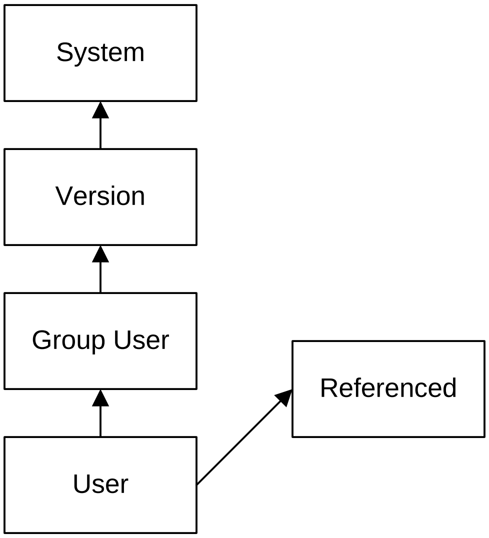


Figure 1-4: Hierarchy for menus and library lists

Each data library in the library list must have the same language code and CCSID.

The S2KRELEASE data area stores the library type.

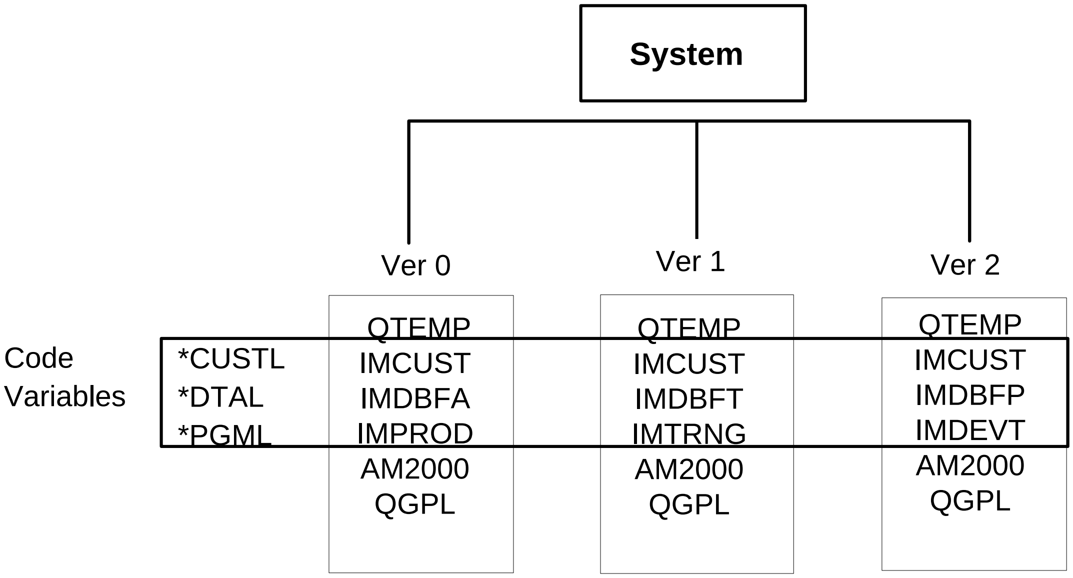
- For data libraries (DTA), the S2KRELEASE data area stores the language and CCSID.

  The system uses the language code and CCSID to determine the language for Infinium applications.

- For program libraries (PGM), the S2KRELEASE data area stores the message files.

Code variables

Code variables allow you to set up a library list or other items with variables to be resolved at run time. The variables can be overridden. For example, if there is a version code variable, it is used in the library list in place of what was there for the system.



*Horizontal box represents code variables.

Figure 1-5: Library list code variables

Job controls

Job controls below allow you to create options that you can put on a menu.

- Text

- Menu
- Program
- Link
- CL or Command

Soft Coder

A method of programming function keys in display files and RPG programs. Having the text for the function keys and the action invoked by the function key “soft coded” as opposed to “hard coded” provides functionality and flexibility.

Entry panels

An entry panel is a prompt screen for any process that requires an input parameter. Entry panels eliminate the coding involved in creating and maintaining the programs that pass parameters to another program. Entry panels provide you with the following:

- A consistent way to design the screen or entry panel
- A consistent way to pass and validate parameters

Prompts

A window of values that can be displayed from fields or menus:

- Design of the window
- Return of a value to the screen

Chapter 2 Job Controls

2

Job control definitions allow you to put different types of jobs on a menu.

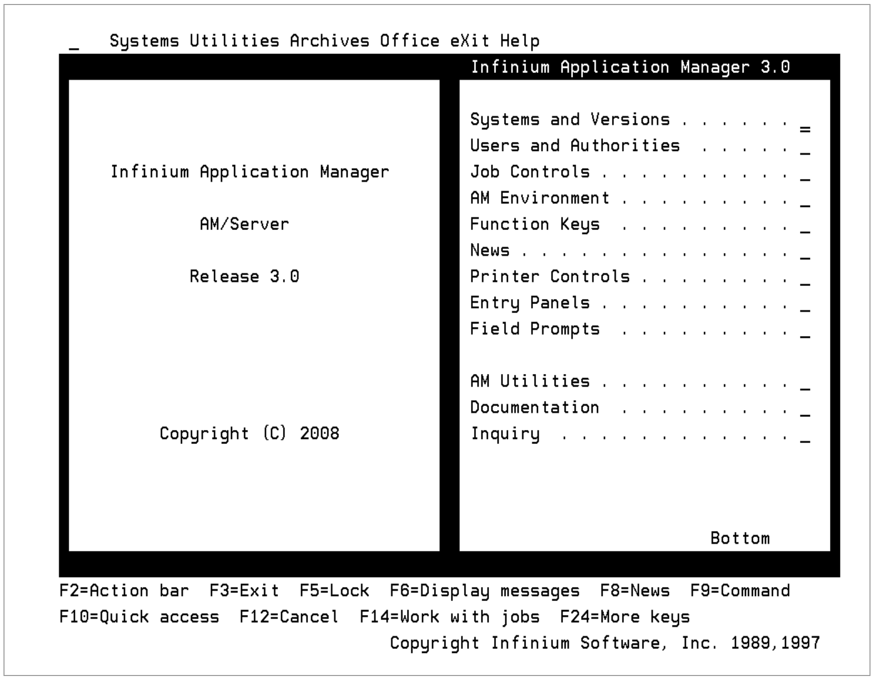
The chapter consists of the following topics:

Job controls overview

What is a job control?

A job control defines an entry on a menu. After selecting a menu item, control is passed to the Event Requester if it is an interactive job and to the Event Manager if it is a batch job as specified in the job control definition. Job controls are:

- Unique by system, release, and modification
- Defined by job type



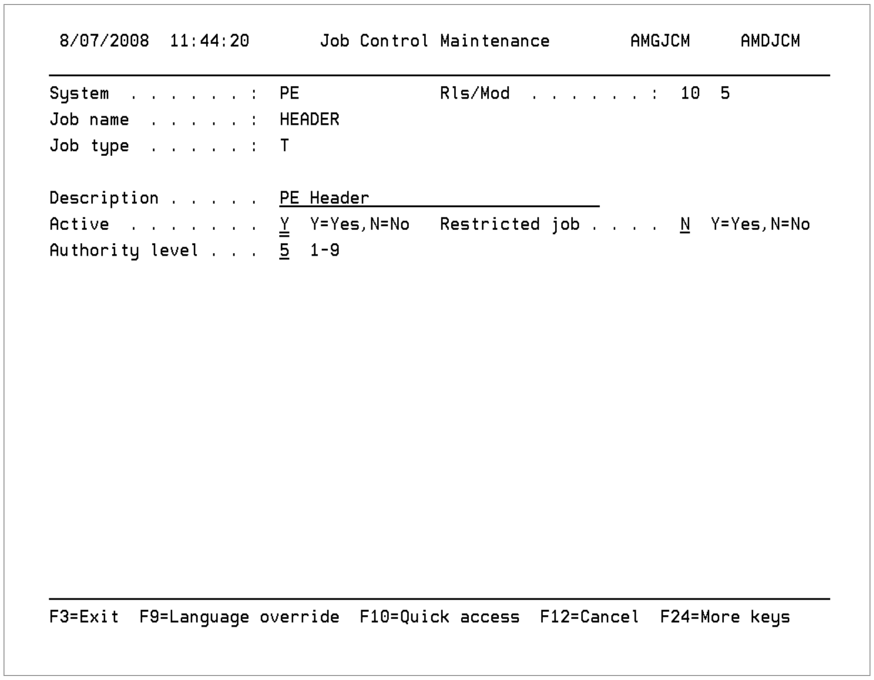
```
 _   Systems Utilities Archives Office eXit Help
                                        Infinium Application Manager 3.0

                                        Systems and Versions . . . . . . =
                                        Users and Authorities  . . . . . _
    Infinium Application Manager        Job Controls . . . . . . . . . . _
                                        AM Environment . . . . . . . . . _
             AM/Server                  Function Keys  . . . . . . . . . _
                                        News . . . . . . . . . . . . . . _
            Release 3.0                 Printer Controls . . . . . . . . _
                                        Entry Panels . . . . . . . . . . _
                                        Field Prompts  . . . . . . . . . _

                                        AM Utilities . . . . . . . . . . _
                                        Documentation  . . . . . . . . . _
         Copyright (C) 2008             Inquiry  . . . . . . . . . . . . _



                                                                Bottom

F2=Action bar  F3=Exit  F5=Lock  F6=Display messages  F8=News  F9=Command
F10=Quick access  F12=Cancel  F14=Work with jobs  F24=More keys
                                 Copyright Infinium Software, Inc. 1989,1997
```

Figure 2-1: Infinium AM main menu screen

Creating a text job control definition

A text (T) job control definition:

- Creates a text entry (blank, subheading, and so on)
- Can be created once and used over and over (overriding on menu)



```
 8/07/2008  11:44:20       Job Control Maintenance        AMGJCM     AMDJCM

System  . . . . . . . :  PE              Rls/Mod  . . . . . . . :  10  5
Job name  . . . . . . :  HEADER
Job type  . . . . . . :  T

Description . . . . .  PE Header
Active  . . . . . . .  Y  Y=Yes,N=No   Restricted job . . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9

F3=Exit  F9=Language override  F10=Quick access  F12=Cancel  F24=More keys
```

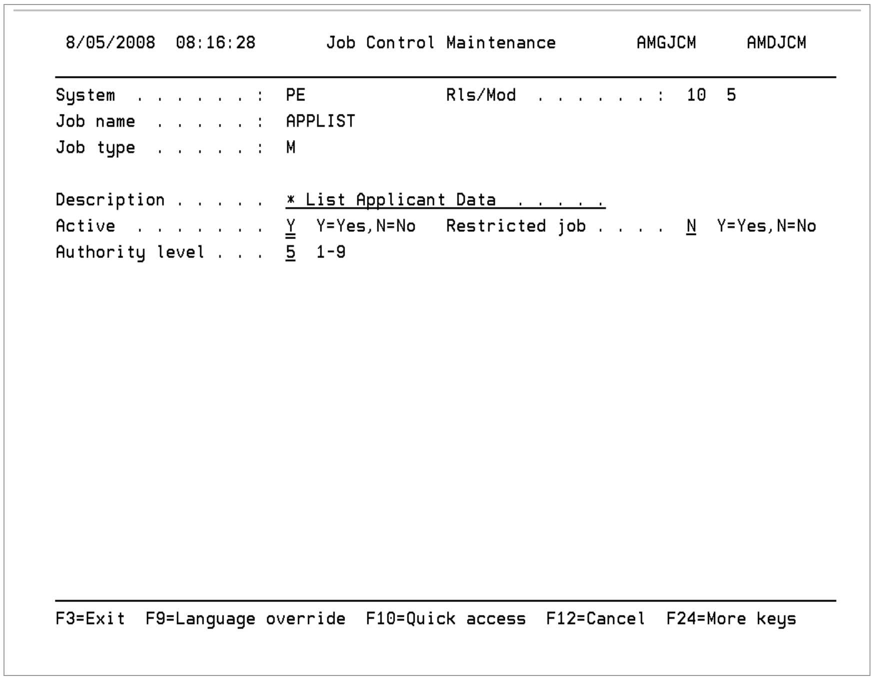
Figure 2-2: Job Control Maintenance screen - text job

Type the description for the text on this screen.

Creating a menu job control definition

A menu (M) job control definition:

- Creates a text entry for a menu
- Can be created once and used over and over (overriding on menu)



```
 8/05/2008  08:16:28       Job Control Maintenance        AMGJCM     AMDJCM

System  . . . . . . :  PE              Rls/Mod  . . . . . . :  10  5
Job name  . . . . . :  APPLIST
Job type  . . . . . :  M

Description . . . . .  * List Applicant Data  . . . . .
Active  . . . . . . .  Y  Y=Yes,N=No   Restricted job . . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9

F3=Exit  F9=Language override  F10=Quick access  F12=Cancel  F24=More keys
```

Figure 2-3: Job Control Maintenance screen - menu job

Type the description for this submenu.

Creating a linking job control definition

A linking (L) job control definition:

- Creates a job control that calls other job controls, which are executed in sequence when the link job control is selected
- Always run interactively although they can spawn batch jobs
- Can link jobs from different systems and versions
- Can link jobs which themselves are link type job controls

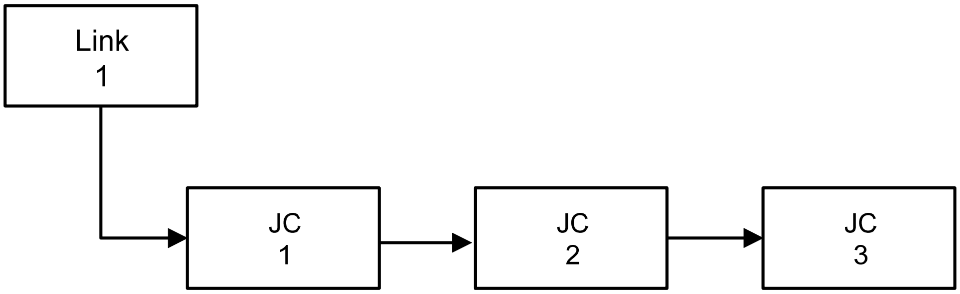
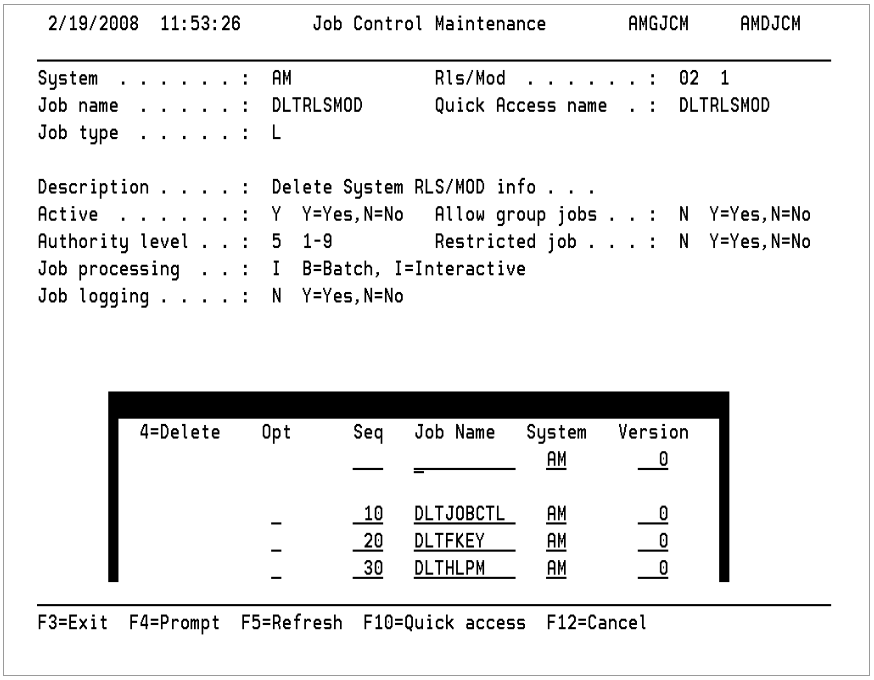


Figure 2-4: Example of linking job control



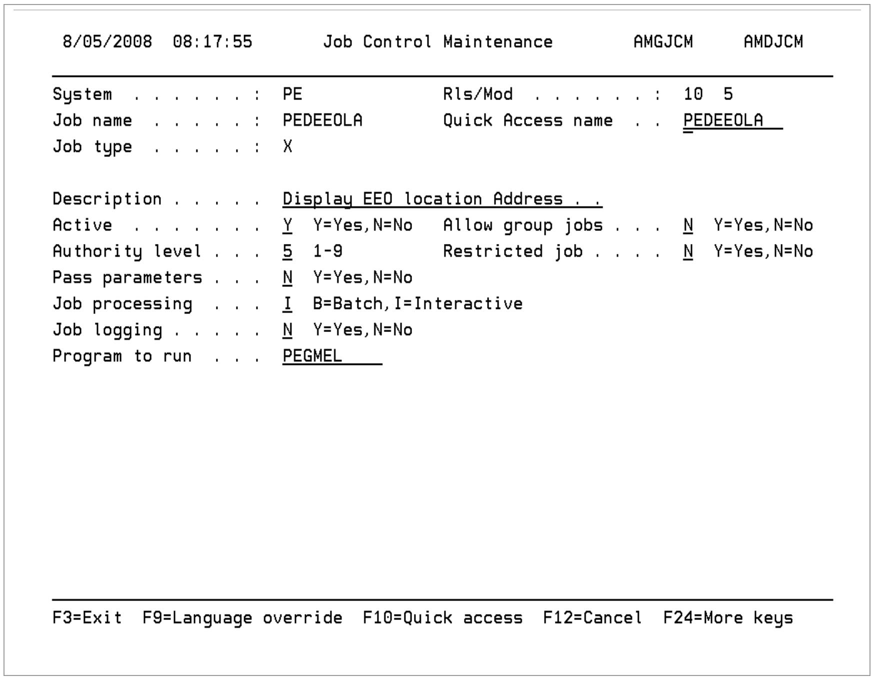
```
 2/19/2008  11:53:26       Job Control Maintenance       AMGJCM     AMDJCM
____________________________________________________________________________
System  . . . . . . :  AM              Rls/Mod  . . . . . . :  02  1
Job name  . . . . . :  DLTRLSMOD       Quick Access name  . :  DLTRLSMOD
Job type  . . . . . :  L

Description . . . . :  Delete System RLS/MOD info . . .
Active  . . . . . . :  Y  Y=Yes,N=No   Allow group jobs . . :  N  Y=Yes,N=No
Authority level . . :  5  1-9          Restricted job . . . :  N  Y=Yes,N=No
Job processing  . . :  I  B=Batch, I=Interactive
Job logging . . . . :  N  Y=Yes,N=No



          4=Delete     Opt       Seq   Job Name    System    Version
                                 ___   _________    AM         __0

                        _        _10   DLTJOBCTL    AM         __0
                        _        _20   DLTFKEY      AM         __0
                        _        _30   DLTHLPM      AM         __0
____________________________________________________________________________
F3=Exit  F4=Prompt  F5=Refresh  F10=Quick access  F12=Cancel
```

Figure 2-5: Job Control Maintenance screen - link job

Creating a program job control definition

You have the following options for a program (X) job control definition:

- Run a batch program
- Run an interactive program
- Run a batch program with parameters (entry panels)
- Run an interactive program with parameters (entry panels)



```
 8/05/2008  08:17:55       Job Control Maintenance        AMGJCM     AMDJCM

System  . . . . . . . :  PE              Rls/Mod  . . . . . . . :  10  5
Job name  . . . . . . :  PEDEEOLA        Quick Access name  . .  PEDEEOLA
Job type  . . . . . . :  X

Description . . . . .  Display EEO location Address . .
Active  . . . . . . .  Y  Y=Yes,N=No   Allow group jobs . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9          Restricted job . . . .  N  Y=Yes,N=No
Pass parameters . . .  N  Y=Yes,N=No
Job processing  . . .  I  B=Batch,I=Interactive
Job logging . . . . .  N  Y=Yes,N=No
Program to run  . . .  PEGMEL

F3=Exit  F9=Language override  F10=Quick access  F12=Cancel  F24=More keys
```

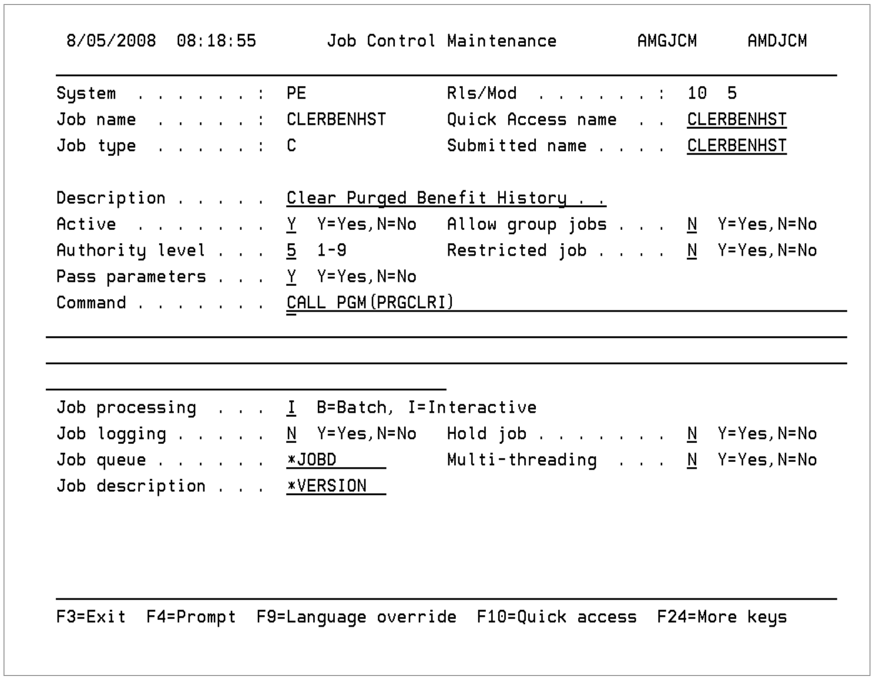
Figure 2-6: Job Control Maintenance screen - program job

Specify an interactive or batch job and the *Pass parameters* flag. Complete the fields that are displayed accordingly.

Creating a command job control definition

You have the following options for a command (C) job control definition:

- Execute a command to run interactively
- Execute a command to run in batch
- Ability to press F4 in the *Command* field to select from a display of valid parameters
- Ability to have a command that prompts the user for parameters by using one of the following methods:
  - By specifying **Y** in the *Pass parameters* field
  - By placing a question mark (**?**) before the command



```
 8/05/2008  08:18:55      Job Control Maintenance       AMGJCM     AMDJCM

System  . . . . . . :  PE              Rls/Mod  . . . . . . :  10  5
Job name  . . . . . :  CLERBENHST      Quick Access name  . .  CLERBENHST
Job type  . . . . . :  C               Submitted name . . . .  CLERBENHST

Description . . . . .  Clear Purged Benefit History . .
Active  . . . . . . .  Y  Y=Yes,N=No   Allow group jobs . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9          Restricted job . . . .  N  Y=Yes,N=No
Pass parameters . . .  Y  Y=Yes,N=No
Command . . . . . . .  CALL PGM(PRGCLRI)



Job processing  . . .  I  B=Batch, I=Interactive
Job logging . . . . .  N  Y=Yes,N=No   Hold job . . . . . . .  N  Y=Yes,N=No
Job queue . . . . . .  *JOBD           Multi-threading  . . .  N  Y=Yes,N=No
Job description . . .  *VERSION




F3=Exit  F4=Prompt  F9=Language override  F10=Quick access  F24=More keys
```

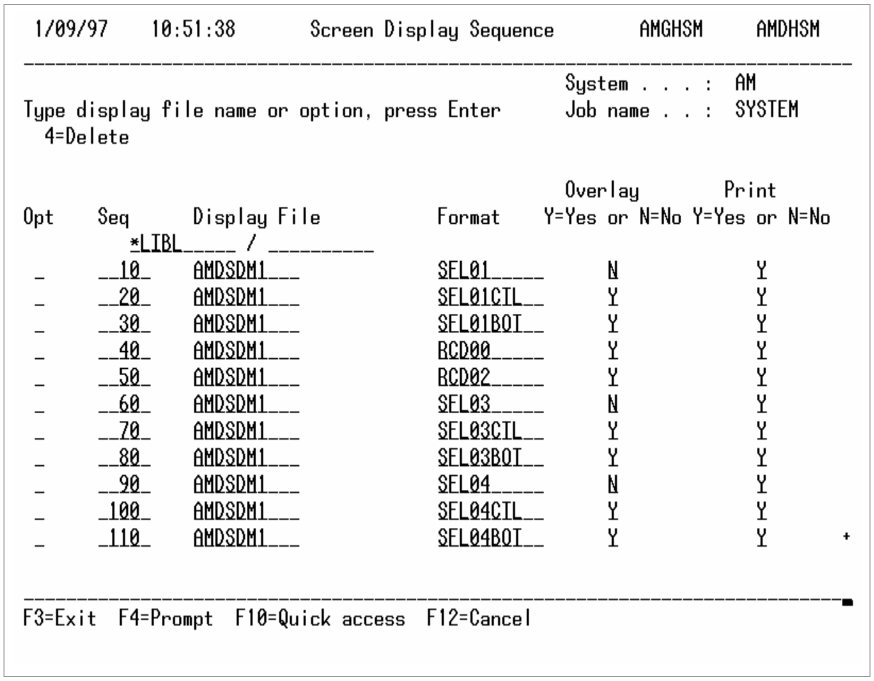
Figure 2-7: Job Control Maintenance screen - command job

Specify an interactive or batch command. Complete the fields that are displayed accordingly.

Specifying related displays

You need to specify related displays F16 for a program (X) job control to ensure that the following will work with your display screens:

- Field level security
- Function level help
- Print help text



```
 1/09/97    10:51:38       Screen Display Sequence        AMGHSM     AMDHSM
 ______________________________________________________________________________
                                                    System . . . :  AM
Type display file name or option, press Enter       Job name . . :  SYSTEM
  4=Delete

                                                    Overlay        Print
Opt    Seq      Display File          Format     Y=Yes or N=No Y=Yes or N=No
          *LIBL_____ / __________
 _     __10_    AMDSDM1___            SEL01_____       N              Y
 _     __20_    AMDSDM1___            SEL01CTL__       Y              Y
 _     __30_    AMDSDM1___            SEL01BOT__       Y              Y
 _     __40_    AMDSDM1___            RCD00_____       Y              Y
 _     __50_    AMDSDM1___            RCD02_____       Y              Y
 _     __60_    AMDSDM1___            SEL03_____       N              Y
 _     __70_    AMDSDM1___            SEL03CTL__       Y              Y
 _     __80_    AMDSDM1___            SEL03BOT__       Y              Y
 _     __90_    AMDSDM1___            SEL04_____       N              Y
 _     _100_    AMDSDM1___            SEL04CTL__       Y              Y
 _     _110_    AMDSDM1___            SEL04BOT__       Y              Y       +

 ______________________________________________________________________________
F3=Exit  F4=Prompt  F10=Quick access  F12=Cancel
```

Figure 2-8: Screen Display Sequence screen

You can specify a job control or ***ALL** for a specific system, release, and modification.

From this screen specify display files associated with the program job control definition.

The utility UPDSCRSEQ is available to populate related display information.

Chapter 3 Soft Coder

3

Soft coded function keys provide you with a method of programming function keys that gives you flexibility and consistency across applications.

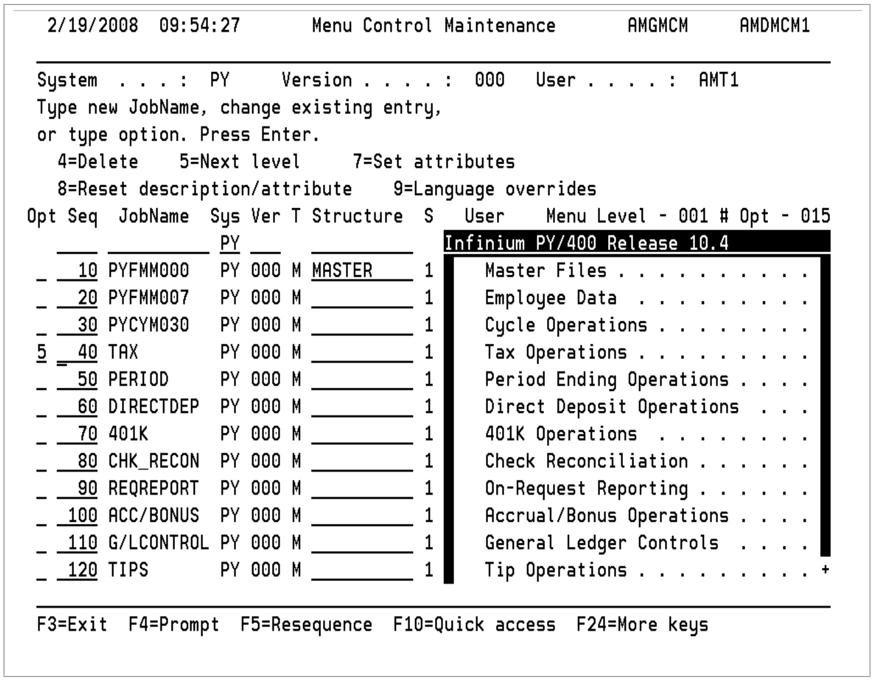
The chapter consists of the following topics:

Overview

Soft Coder is a technique used to code function keys in display files and RPG programs. It provides you with an easy way to quickly change function key definitions in an application.

Soft Coder:

- Is unique by system, release, and modification
- Is activated by completing *Function Key Maintenance* in Infinium AM
- Requires coding in the display file and RPG program

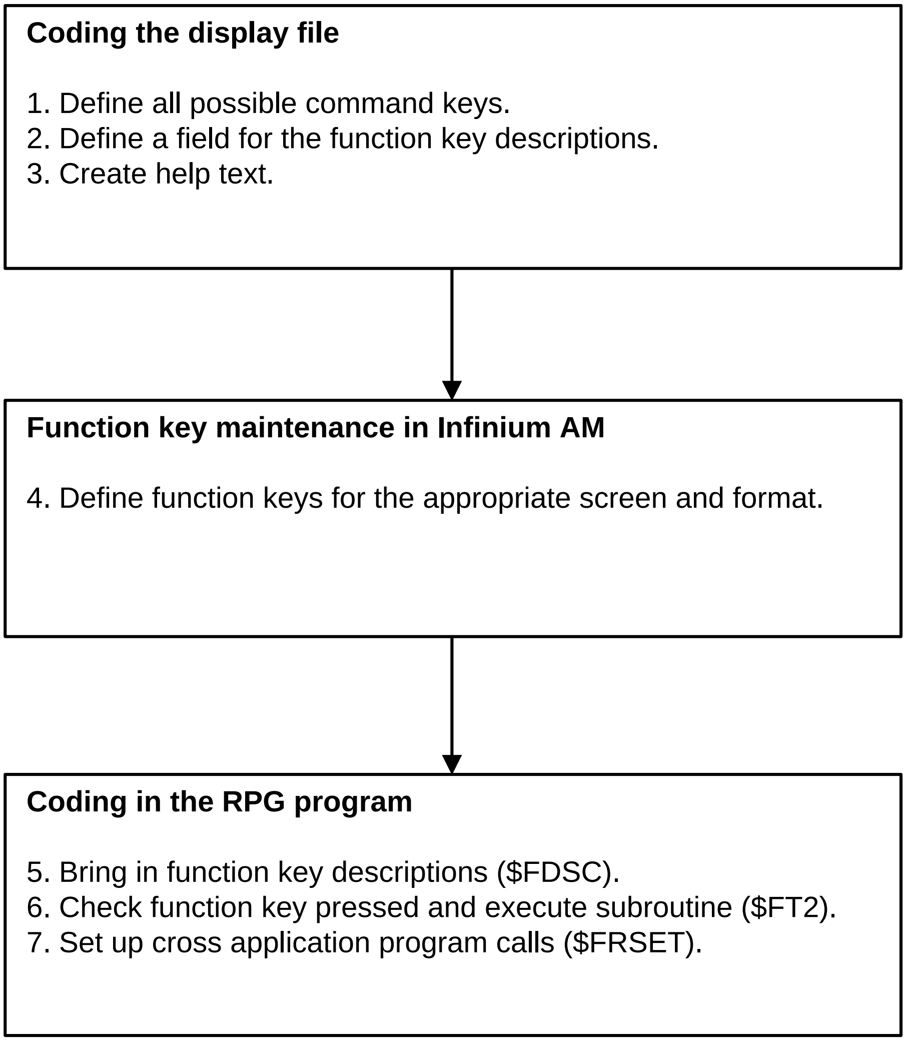


```
 2/19/2008  09:54:27       Menu Control Maintenance       AMGMCM     AMDMCM1

System  . . . :  PY     Version . . . . :  000   User . . . . :  AMT1
Type new JobName, change existing entry,
or type option. Press Enter.
  4=Delete    5=Next level     7=Set attributes
  8=Reset description/attribute    9=Language overrides
Opt Seq  JobName  Sys Ver T Structure  S   User    Menu Level - 001 # Opt - 015
   ____ __________ PY ___   __________    Infinium PY/400 Release 10.4
_  __10 PYFMM000   PY 000 M MASTER____ 1     Master Files . . . . . . . . . .
_  __20 PYFMM007   PY 000 M __________ 1     Employee Data  . . . . . . . . .
_  __30 PYCYM030   PY 000 M __________ 1     Cycle Operations . . . . . . . .
5  __40 TAX        PY 000 M __________ 1     Tax Operations . . . . . . . . .
_  __50 PERIOD     PY 000 M __________ 1     Period Ending Operations . . . .
_  __60 DIRECTDEP  PY 000 M __________ 1     Direct Deposit Operations  . . .
_  __70 401K       PY 000 M __________ 1     401K Operations  . . . . . . . .
_  __80 CHK_RECON  PY 000 M __________ 1     Check Reconciliation . . . . . .
_  __90 REQREPORT  PY 000 M __________ 1     On-Request Reporting . . . . . .
_  _100 ACC/BONUS  PY 000 M __________ 1     Accrual/Bonus Operations . . . .
_  _110 G/LCONTROL PY 000 M __________ 1     General Ledger Controls  . . . .
_  _120 TIPS       PY 000 M __________ 1     Tip Operations . . . . . . . . . +

F3=Exit  F4=Prompt  F5=Resequence  F10=Quick access  F24=More keys
```

Figure 3-1: Soft Coder - what the user sees

The user sees consistent function keys at the bottom of each screen.

Steps to implement



Coding the display file

1. Define all possible command keys.
2. Define a field for the function key descriptions.
3. Create help text.

Function key maintenance in Infinium AM

4. Define function keys for the appropriate screen and format.

Coding in the RPG program

5. Bring in function key descriptions ($FDSC).
6. Check function key pressed and execute subroutine ($FT2).
7. Set up cross application program calls ($FRSET).

Figure 3-2: Soft Coder implementation steps

Coding in the display file

Step 1: Define the command keys

Define all possible command keys as follows:

- Code command keys above the first record format.
- Code all possible function keys including Enter, Help, Page Up, Page Down, and so on.
- Define keys as a command function.

```
A*%%EC
A                                      DSPSIZ(24 80 *DS3)
A                                      REF(*LIBL/ADFLDREF)
A                                      MSGLOC(24)
A                                      PRINT(*LIBL/PRINTKEY)
A                                      HELP
A                                      CF01
A                                      CF02
A                                      CF03
A                                      CF04
A                                      CF05
A                                      CF06
A                                      CF07
A                                      CF08
A                                      CF09
A                                      CF10
A                                      CF11
A                                      CF12
A                                      CF13
A                                      CF14
A                                      CF15
A                                      CF16
A                                      CF17
A                                      CF18
A                                      CF19
A                                      CF20
A                                      CF21
A                                      CF22
A                                      CF23
A                                      CF24
A                                      CLEAR
A                                      HOME
A                                      ROLLUP
A                                      ROLLDOWN
A* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
A          R RCD01
A*%%TS  SD  19940725  164536  PARE        REL-V2R3M0  5738-PW1
```

Figure 3-3: Sample code for defining command keys

Step 2: Define a field for function key descriptions

Define a field for function key descriptions for one of the following three formats:

- Full screen format
  - Record format line 23 (Line 22 for 2 lines of keys)
  - Output field
  - Starting at position 2
  - Length of 78 characters
- Subfile format
  - Defined in a format associated with the subfile, such as SFLBOT
- Window format
  - Defined for bottom line of window
  - Length of window determines length of field

```
A            R1LOC     R        B  9 30REFFLD(FLLOC)
A                                 22  2'                                     -
A                                                                            -
A                                              '
A                                      COLOR(BLU)
A                                      DSPATR(UL)
A**
A**  Insert an output field for function key descriptions.
A**
A            FK01          78A  O 23  2COLOR(BLU)
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 3-4: Defining a field for function key descriptions - main screen

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
A          R SFLBOT01
A*%%TS  SD  19940720  153227  PARE        REL-V2R3M0  5738-PW1
A                                      OVERLAY
A                                 22  2'                                     -
A                                                                            -
A                                              '
A                                      COLOR(BLU)
A                                      DSPATR(UL)
A**
A**  Insert an output field for function key descriptions.
A**
A            FKSF          78A  O 23  2COLOR(BLU)
```

Figure 3-5: Defining a field for function key descriptions - subfile

Step 3: Create help text

After you complete coding in the display file, use one of the following methods to create help text:

- Compile the display file and run the **CRTHLPTXT** command
- Run the **AM2000/CRTDSPF** command

Function key maintenance

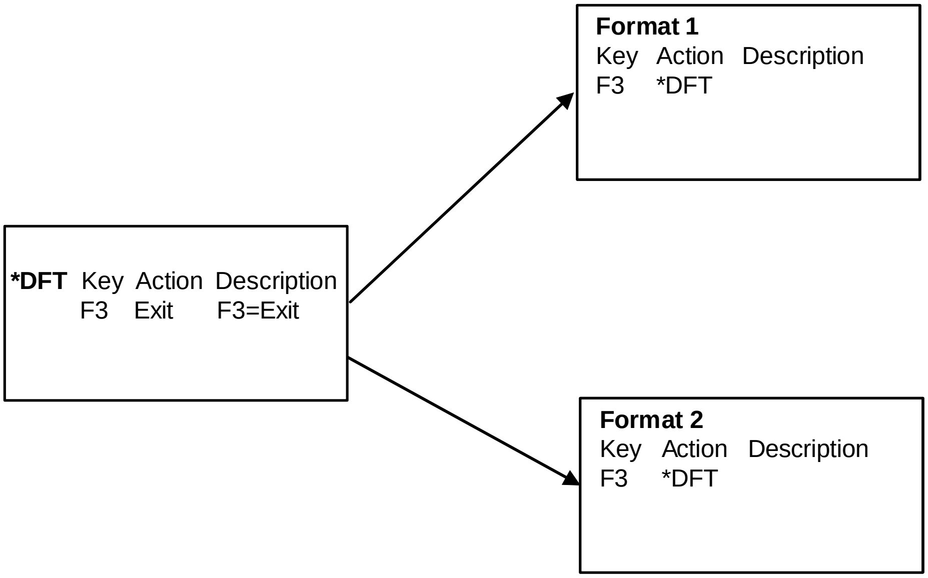
Step 4: Define function keys for the screen and format

You need to perform function key maintenance so that you can maintain the keys for each display file format. The types of function key definitions are:

- *DFT
- *MENU
- *PROMPT
- Record formats

Using *DFT to define keys

The diagram in Figure 3-6: depicts how you can set up the *DFT key and use it in different formats.

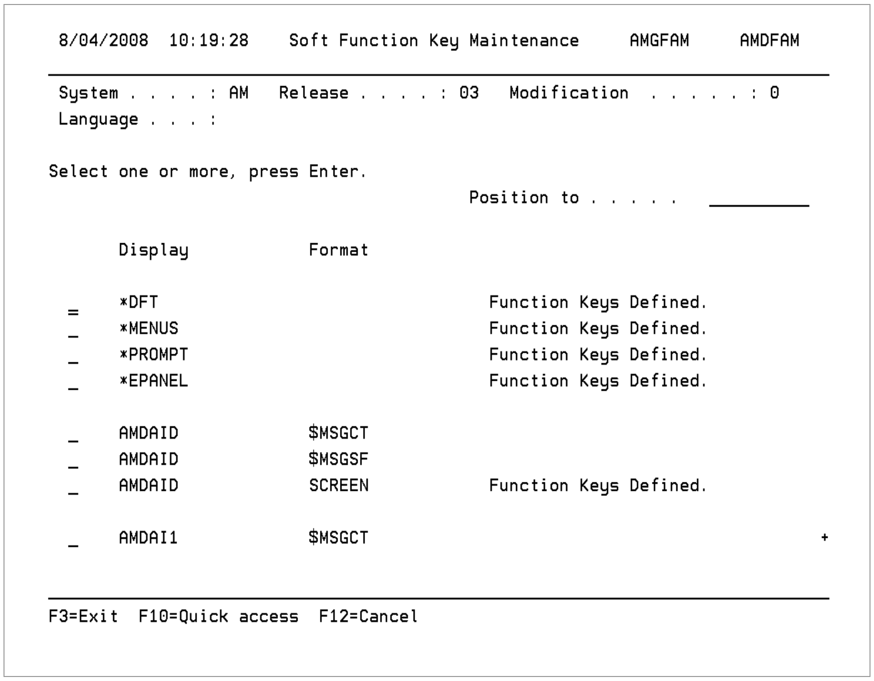


Format 1
Key Action Description
F3 *DFT

***DFT** Key Action Description
F3 Exit F3=Exit

Format 2
Key Action Description
F3 *DFT

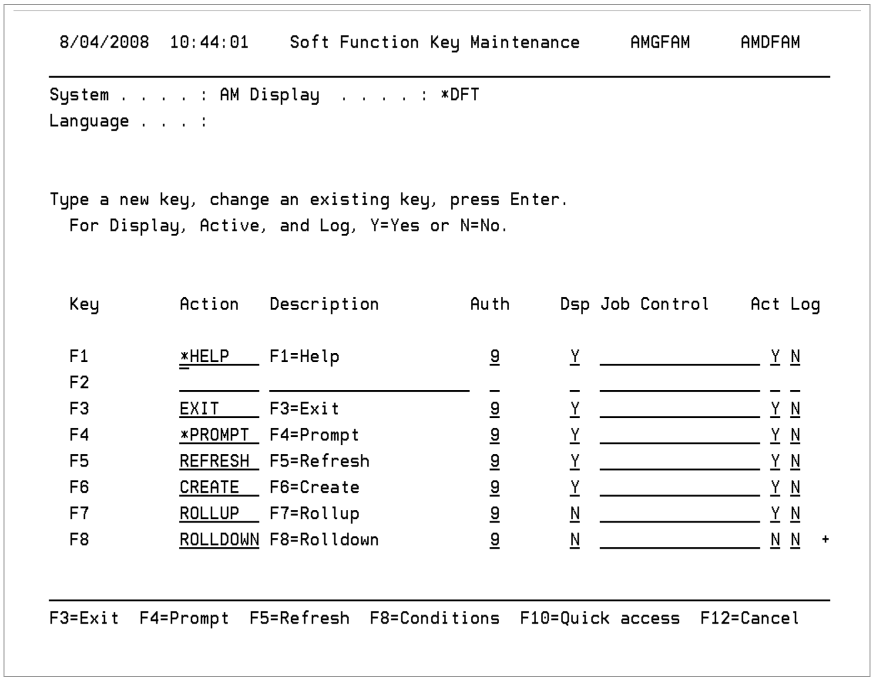
Figure 3-6: Using the *DFT key in two formats



```
 8/04/2008  10:19:28    Soft Function Key Maintenance     AMGFAM     AMDFAM
______________________________________________________________________________
 System . . . . : AM    Release . . . . : 03   Modification  . . . . . : 0
 Language . . . :

Select one or more, press Enter.
                                          Position to . . . . .   __________

       Display           Format

  =    *DFT                                 Function Keys Defined.
  _    *MENUS                               Function Keys Defined.
  _    *PROMPT                              Function Keys Defined.
  _    *EPANEL                              Function Keys Defined.

  _    AMDAID            $MSGCT
  _    AMDAID            $MSGSF
  _    AMDAID            SCREEN             Function Keys Defined.

  _    AMDAI1            $MSGCT                                            +

______________________________________________________________________________
F3=Exit  F10=Quick access  F12=Cancel
```

Figure 3-7: Soft Function Key Maintenance - display selection screen

From this screen specify the record format for which you are going to define function keys. Define function keys for subfiles at the control file. You can use a utility to copy keys from system to system.



```
 8/04/2008  10:44:01    Soft Function Key Maintenance     AMGFAM     AMDFAM
______________________________________________________________________________
System . . . . : AM Display  . . . . : *DFT
Language . . . :


Type a new key, change an existing key, press Enter.
  For Display, Active, and Log, Y=Yes or N=No.


  Key        Action   Description         Auth     Dsp Job Control    Act Log

  F1         *HELP    F1=Help               9        Y  ________________  Y N
  F2         ________ ____________________  _        _  ________________  _ _
  F3         EXIT     F3=Exit               9        Y  ________________  Y N
  F4         *PROMPT  F4=Prompt             9        Y  ________________  Y N
  F5         REFRESH  F5=Refresh            9        Y  ________________  Y N
  F6         CREATE   F6=Create             9        Y  ________________  Y N
  F7         ROLLUP   F7=Rollup             9        N  ________________  Y N
  F8         ROLLDOWN F8=Rolldown           9        N  ________________  N N  +

______________________________________________________________________________
F3=Exit  F4=Prompt  F5=Refresh  F8=Conditions  F10=Quick access  F12=Cancel
```

Figure 3-8: Soft Function Key Maintenance - function key maintenance

Coding in the RPG program

Step 5: Bring in function key descriptions

Follow these steps as part of the initialization subroutine. Do this only once in each program:

1. Perform a move left on the display file to field STDFIL.
2. Move the first record format to field STDFMT.
3. Execute subroutine $FDSC.
4. Perform a move left on the function key descriptions to the field defined in the display file.

Repeat the above for all record formats in the display file.

```
 *1
 *1   SETUP INITIAL VALUES FOR FUNCTION KEYS
 *1
 *1   Replace these Key intializations with the subroutine
 *1   to return function key descriptions ($FDSC)
 *1
C                     MOVEL'ADDWINVS'STDFIL
C                     MOVEL'RCD01'   STDFMT
C                     EXSR $FDSC
C                     MOVEL$DSC      FK01
 *
```

Figure 3-9: Bringing in function key descriptions - main screen

The above code initializes a display file and a record format and then calls AMIFD in order to bring in the function key descriptions.

Step 6: Validate function key pressed and execute subroutine

The following code validates which function key is pressed. Code must be added for all record formats and subfiles to perform the following:

- Read the display record format to determine which function key is pressed.

- Execute the $FT2 subroutine, which returns the function key action and handles any action with an asterisk in front of it in the following order (AMIFT2):

1. *PROMPT
2. *HELP
3. *JOBCTL
4. *MORKYS
5. *NODEF

```
C           CURLVL    DOWEQPRCLVL
 *
C                     WRITE$MSGCT
 *1
 *1 Add the code to bypass the WRITE opcode if HELP is pressed.
 *1 Add the code to that executes subroutine $FT2.
 *1
C           $ACTT     IFNE 'A'
C                     WRITERCD01
C                     ENDIF
C                     READ RCD01                    99
 *
C                     MOVE 'Y'       $ERRIN
C                     EXSR $ERROR
 *
C                     EXSR $FT2
 *1
 *1 The following hard coded key actions
 *1 need to be changed to use Soft Coder.
 *1
C           $ACTN     CASEQ'EXIT'    EXIT
C           $ACTN     CASEQ'DSPVND'  DSPVND
C           $ACTN     CASEQ'RETURN'  EXIT
C           $ACTN     CASEQ'ENTER'   PROC02
C           $ACTN     CASEQ'*PROMPT' PRMPT
C           $ACTN     CASEQ'*HELP'   $ALIGN
C           $ACTN     CASEQ'*MORKYS' MORKYS
C                     ENDCS
 *
C                     ENDDO
 *
```

Figure 3-10: Validating the function key pressed

Step 7: Set up cross-application program calls

If calls are made to programs in other soft coded applications, you need to set the function keys to the cross-application environment prior to calling the cross-application program.

- Prior to call:
    - Specify the system designator of the cross application.
    - Perform a move left of the display file to STDFIL.
    - Execute subroutine $FSET.
- After the call, execute subroutine $FRSET.

The following code shows the call to cross-application program CXGDVND.

```
C           DSPVND    BEGSR
 *
C                     MOVELR1VEND    VENDOR
 *
C                     MOVE STDFIL    SAVFIL
C                     MOVEL'CXDDVND 'STDFIL
C                     MOVE 'CX'      $SYSD
C                     EXSR $FSET
 *
C                     CALL 'CXGDVND'
C                     PARM           VENDOR
 *
C                     EXSR $FRSET
C                     MOVE SAVFIL    STDFIL
```

Figure 3-11: Setting up cross-application program calls

The above code specifies the system designator of the cross application and sets the function key environment. After returning from the program, the function key environment is returned to the original application.

Notes

Chapter 4 Event Processing

4

In this chapter you learn the basic flow of event processing within Infinium AM.

The chapter consists of the following topics:

Overview

The diagram in Figure 4-1 illustrates the event processing flow of data.

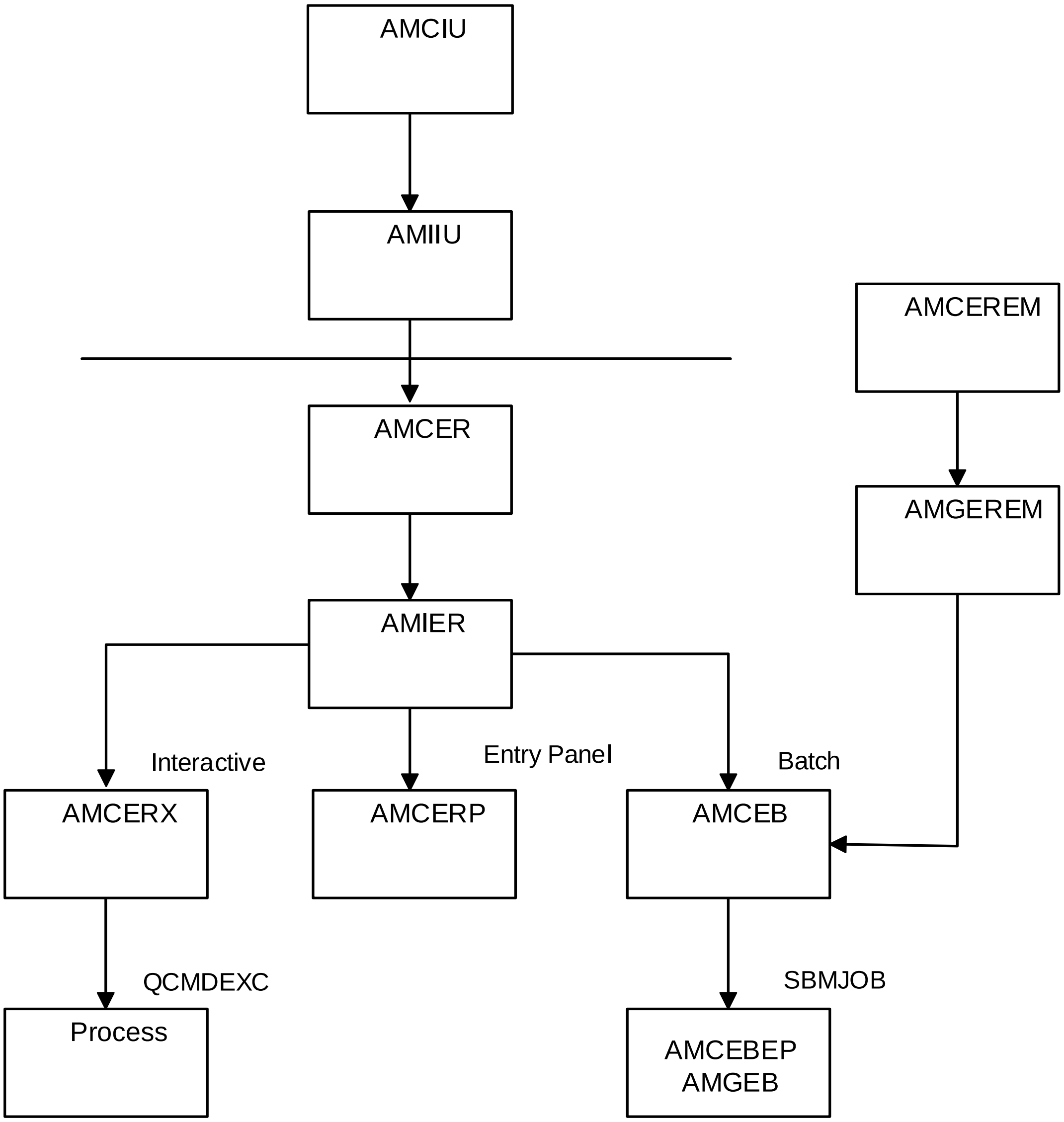


Figure 4-1: Event processing flow

Interactive processing

AMCIU - Initial program

AMCIU is the initial program for all Infinium applications. This program:

- Sets up the library list
- Calls *INITIAL programs
- Opens some common Infinium AM files
- Changes the job to LOGCLPGM(*NO)
- Sets the user or job message queue to use AMCCBRKH
- Calls AMIIU

AMIIU - Main Infinium AM processing program

AMIIU is the main Infinium AM processing program. This program:

- Builds menu panels
- Processes menu function keys
- Processes action bar requests
- Determines what job control to execute when a menu option is selected

AMCER - Event requester program

AMCER is the event requester program. This program:

- Retrieves current job information
- Is the command processing program for RQSAMJOB

AMIER - Main event requester program

AMIER is the main event requester program. This program:

- Determines job control parameters and processes the job based on them
- Builds a 28 byte parameter key for task coupling record
- Calls entry panel processing program
- Calls AMCERX to execute an interactive job control or AMCEB to submit the job to batch
- Builds the command to call the batch driver, usually AMGEB, if necessary

AMCERX - Event requester command processor program

AMCERX is the event requester command processor program. This program calls the command processing program to execute command or call program.

AMCERP - Event requester prompt caller program

AMCERP is the event requester prompt caller program. This program calls entry panel program of Infinium AM’s entry panel processor.

RQSAMJOB - Request Infinium AM job

The RQSAMJOB command is used to execute a job control from a command line or a non-Infinium menu. The Infinium AM library must be in the library list and the user must be authorized to the system version. Parameters are:

System	2 characters
Version	3,0 decimals
Job Control	10 characters

Batch processing

AMCEB - Batch job submission program

AMCEB is the batch job submission program. This program:

- Determines whether to send messages to the job or the user message queue
- Resolves job queues and the job description
- Duplicates the job description in QTEMP
- Modifies the library list (uses current interactive list)
- Modifies user and hold parameters on the job description
- Modifies the job queue parameter, if one was resolved
- Submits job to batch using the job description in QTEMP
- Deletes job description from QTEMP after job has been submitted

Note: Command for submitted job is always a call to the batch driver regardless of C or X job type.

AMCEBEP - Batch driver program

AMCEBEP is the batch driver program. This program:

- Determines the language overrides
- Calls AMGEB

AMGEB - Batch driver program

AMGEB is the batch driver program. This program:

- Is the initial program called in batch
- Determines job run requirements such as archive file, overrides printer files, and calls programs

Alternative job submission

AMCEREM - Alternate batch submission program

AMCEREM is the alternate batch submission program. This is the command processing program for SBMAMJOB. This program sets up the Infinium AM environment.

AMGEREM - Alternate batch submission program

AMGEREM is the alternate batch submission program. This program determines the job control type and calls AMCEB to submit the job.

SBMAMJOB - Submit Infinium AM job

The SBMAMJOB command is used to submit a job control to batch from a command line or non-Infinium menu. The Infinium AM library must be in the library list and the user must be authorized to the system version.

Parameters are:

System	2 characters
Version	3,0 decimals
Job Control	10 characters
28 byte key	28 characters

Alternate batch job submission requirements are:

- Create a batch job control with the following field values:

Yes	*Pass parameters*
***DUMMY**	*Entry panel*

- Create task coupling key and write record to task coupling file

- Call AMCEREM or execute SBMAMJOB using the batch job name and task coupling key

Notes

Chapter 5 Help

5

When you complete this chapter, you will know how to populate the help files and how to call help from a program

The chapter consists of the following topics:

Overview

The following help is available

- Interface help: from a non-functional area on the main menu
- Function help: from a menu option line of a menu
- Field help: from an input field on a screen display
- Screen help: from a non-functional area on a screen display
- Extended help: by pressing F2 any time after pressing Help

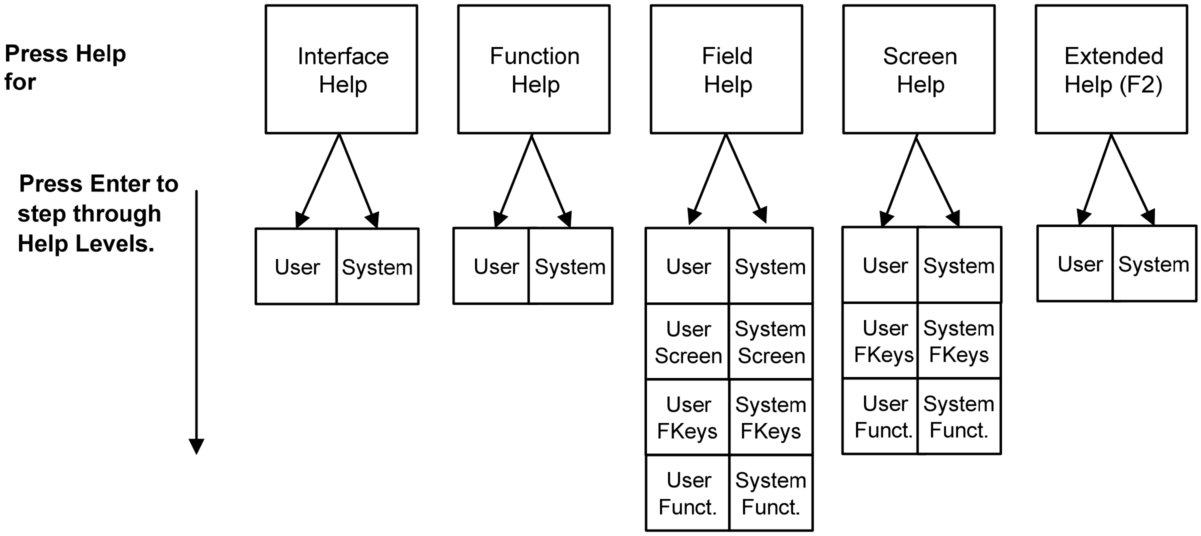


Figure 5-1: Help flow

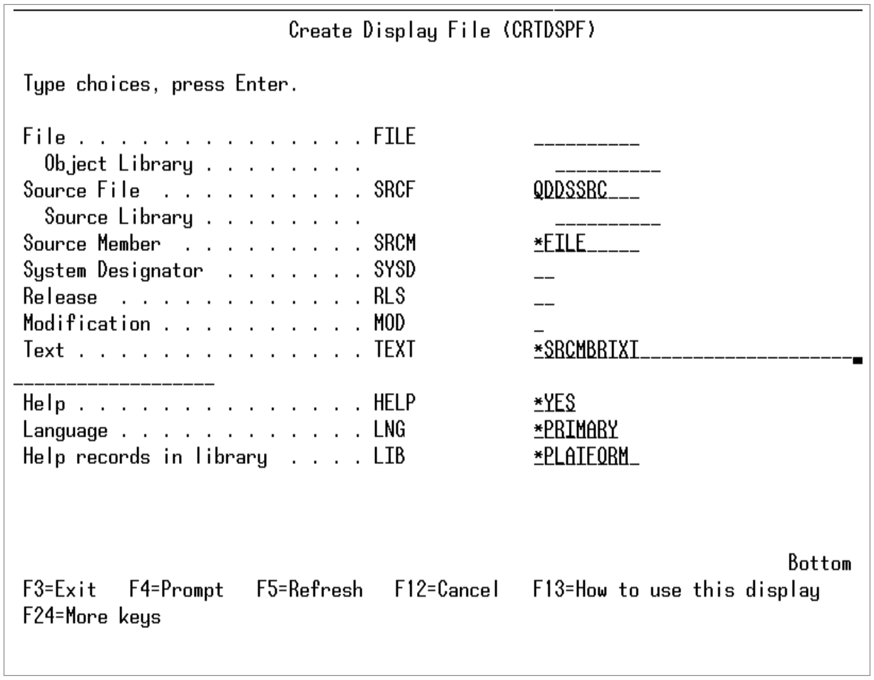
Help files

You can populate help files by running one of the following commands:

- The AM2000/CRTDSPF command creates a new object in the specified library and populates the help files.
- The CRTHLPTXT command only populates the help files.

The following help files are populated:

- AMPHP - Primary controlling file
- AMPHL - Linkage controlling file
- AMPHC - Screen control file
- AMPHK1 - User function key file
- AMPHK2 - System function key file



```
                        Create Display File (CRTDSPF)

 Type choices, press Enter.

 File . . . . . . . . . . . . . . FILE           __________
   Object Library . . . . . . . .                  __________
 Source File  . . . . . . . . . . SRCF           QDDSSRC___
   Source Library . . . . . . . .                  __________
 Source Member  . . . . . . . . . SRCM           *FILE_____
 System Designator  . . . . . . . SYSD           __
 Release  . . . . . . . . . . . . RLS            __
 Modification . . . . . . . . . . MOD            _
 Text . . . . . . . . . . . . . . TEXT           *SRCMBRTXT____________________
___________________
 Help . . . . . . . . . . . . . . HELP           *YES
 Language . . . . . . . . . . . . LNG            *PRIMARY
 Help records in library  . . . . LIB            *PLATFORM_



                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 5-2: AM2000 Create Display File (CRTDSPF) screen

Other help files are:

- AMPHT1 - User text file
- AMPHT2 - System text file

- AMPHS - Screen sequencing and related displays file

Implementing help

You can implement help in one of the following ways:

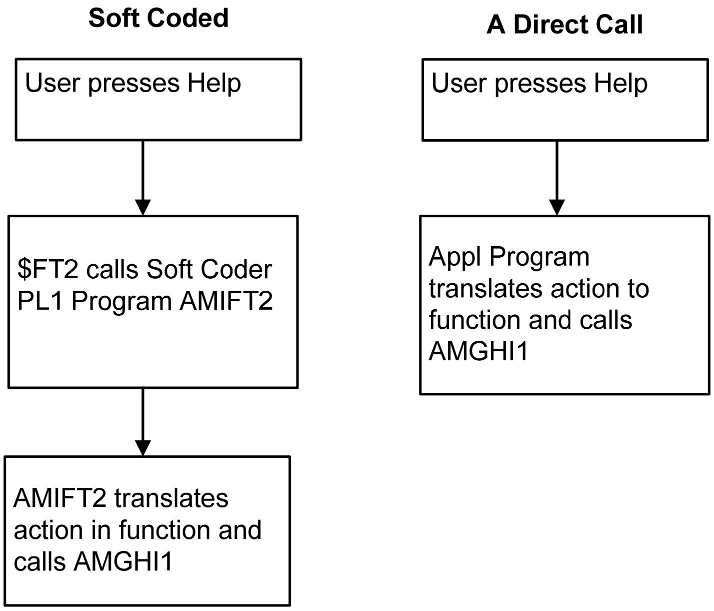

Soft Coded
A Direct Call
User presses Help
User presses Help
$FT2 calls Soft Coder PL1 Program AMIFT2
Appl Program translates action to function and calls AMGHI1
AMIFT2 translates action in function and calls AMGHI1

Figure 5-3: Methods for implementing help

You need to add the code below to have the Soft Coder handle the call to help.

```
C                    EXSR $FT2
 *1
 *1 The following hard coded key actions
 *1 need to be changed to use Soft Coder.
 *1
C          $ACTN     CASEQ'EXIT'    EXIT
C          $ACTN     CASEQ'DSPVND'  DSPVND
C          $ACTN     CASEQ'RETURN'  EXIT
C          $ACTN     CASEQ'ENTER'   PROC02
C          $ACTN     CASEQ'*PROMPT' PRMPT
C          $ACTN     CASEQ'*HELP'   $ALIGN
C          $ACTN     CASEQ'*MORKYS' MORKYS
C                    ENDCS
```

Figure 5-4: Sample code in application

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C          $ALIGN    BEGSR
 *
```

```
10600     C                    MOVE '1'      *IN48
10700      *
10800     C                    ENDSR
10900      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 5-5: $ALIGN subroutine in $AMSTD

Chapter 6 Entry Panels

6

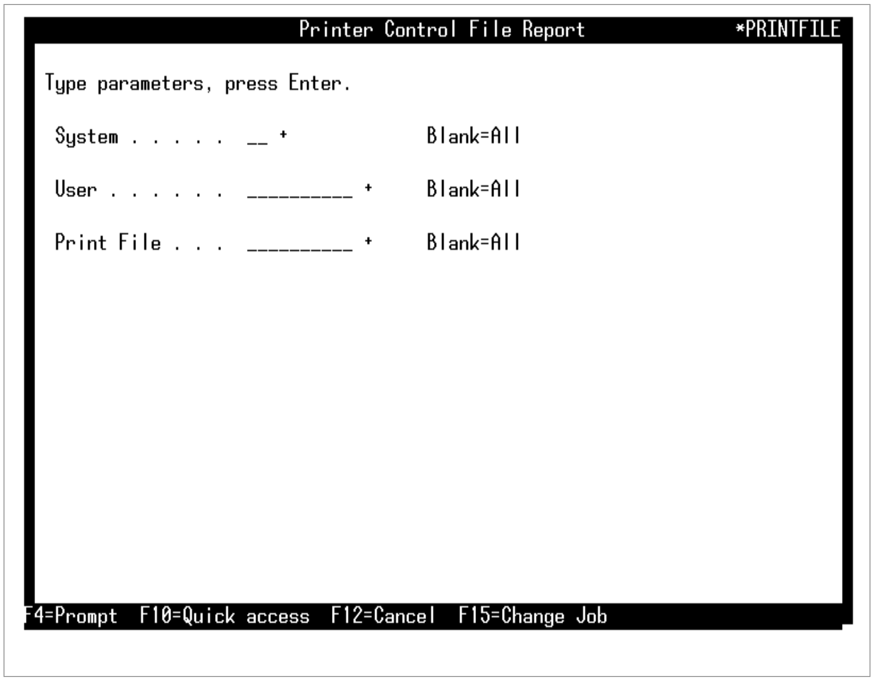
When you complete this chapter, you should know how to define entry panels with Infinium AM and activate an entry panel by modifying the actual code.

The chapter consists of the following topics:

Overview

An entry panel is a prompt screen for any process that requires a parameter input. An entry panel definition is made up of two parts:

- The design of the entry panel
- Field validity checking for the input



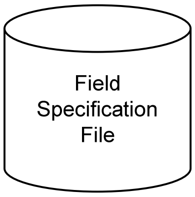
```
                    Printer Control File Report              *PRINTFILE

Type parameters, press Enter.

 System . . . . .  __ +              Blank=All

 User . . . . . .  __________ +      Blank=All

 Print File . . .  __________ +      Blank=All

F4=Prompt  F10=Quick access  F12=Cancel  F15=Change Job
```

Figure 6-1: Example of an entry panel

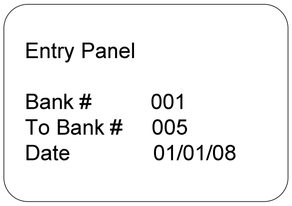
Creating an entry panel

There are three steps involved in creating an entry panel:

1. Creating the necessary field specification files one time for the system.



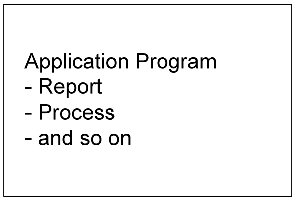
2 Designing the panel and defining validity checking (optional) for each field.



Entry Panel

Bank #	001
To Bank #	005
Date	01/01/08

3 Adding code to the application program that is going to use the parameters as input.

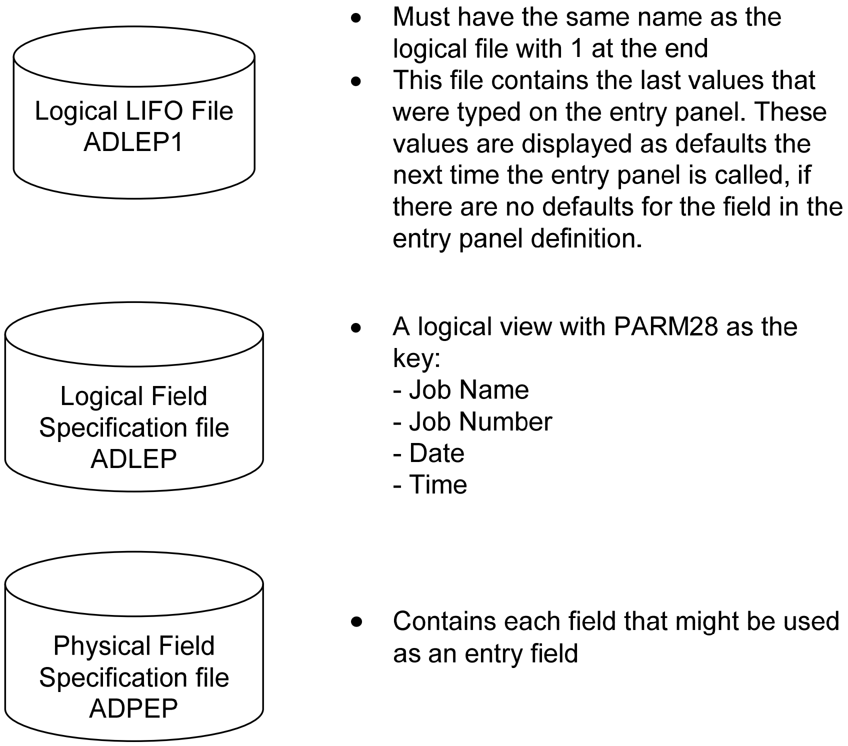


Application Program
- Report
- Process
- and so on

Coding for entry panels - creating files

Field specification files

Setting up the field specification file is usually a one time setup for an application. Create the files with fields that you anticipate being used for entry panels. There is one physical file to be built with two logical files built over that physical file.



- Must have the same name as the logical file with 1 at the end
- This file contains the last values that were typed on the entry panel. These values are displayed as defaults the next time the entry panel is called, if there are no defaults for the field in the entry panel definition.

- A logical view with PARM28 as the key:
  - Job Name
  - Job Number
  - Date
  - Time

- Contains each field that might be used as an entry field

Figure 6-2: Field specification files

Coding for field specification files

Sample ADPEP

```
           R ADREP
             EPPARM    R               TEXT('28 CHAR KEY') 29
             EPCUSR    R               TEXT('CURRENT USER')
             EPWRHS    R               TEXT('Warehouse . . . .')
             EPTWRH    R               TEXT('To Warehouse  . .')
 *
 * Add more fields if you anticipate implementing more entry
 * panels.
```

Sample ADLEP

```
 *
A          R ADREP                     PFILE(ADPEP)
A          K EPPARM
```

Sample ADLEP1

```
 *
A                                      LIFO
A          R ADREP                     PFILE(ADPEP)
 *
 * The two fields EPPARM and EPCUSR must be added here.
 *
A            EPPARM
A            EPCUSR
 *
 * Below are the fields that will be defined in the entry panels.
 *
A            EPWRHS
A            EPTWRH
 *
 *
 * (Add more fields if you anticipate implementing more entry
 * panels.)
 *
 * The fields below MUST be added last in the LIFO file (do not
 * add them to the physical file).  Note that the field JOBNAM
 * is declared as a substring--SST--of EPPARM occupying the 1st
 * 10 bytes of EPPARM's 28 byte key.
 *
 *
A            JOBNAM             I      SST(EPPARM 1 10)
A          K JOBNAM
           K EPCUSR
```

Entry panel processing flow

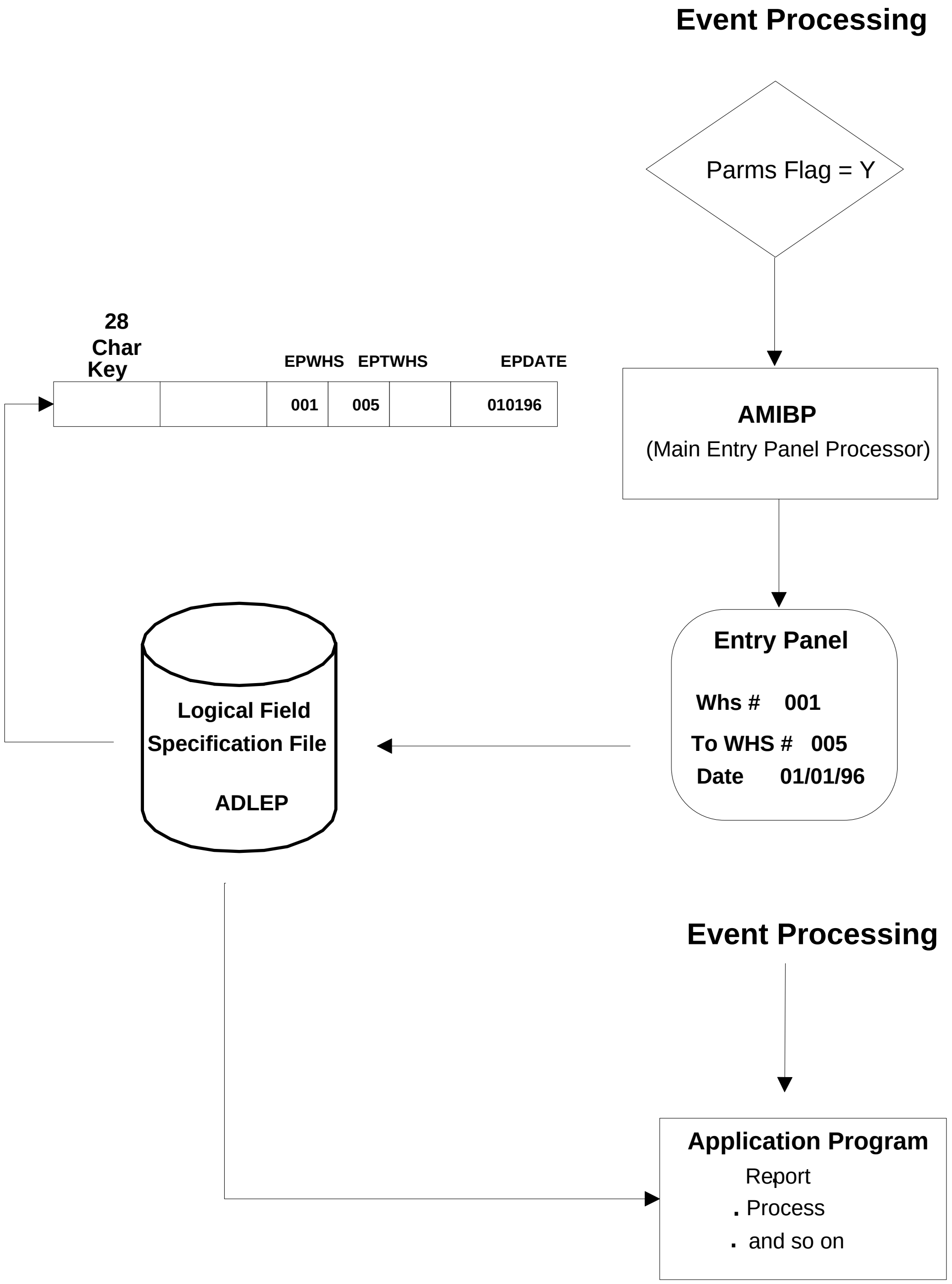
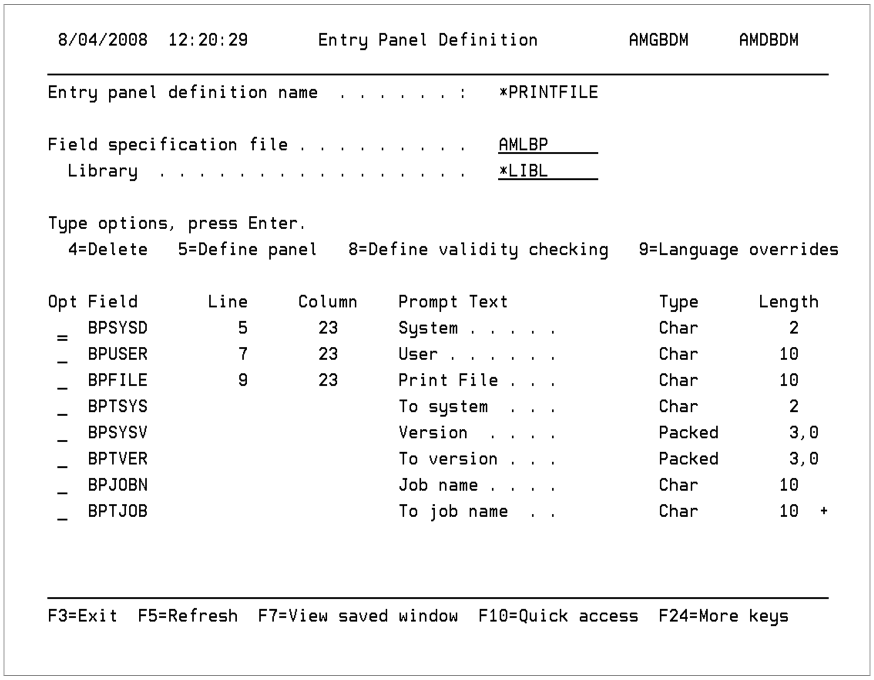


Figure 6-3: Entry panel processing flow

Designing the entry panel

The components to an entry panel definition are:

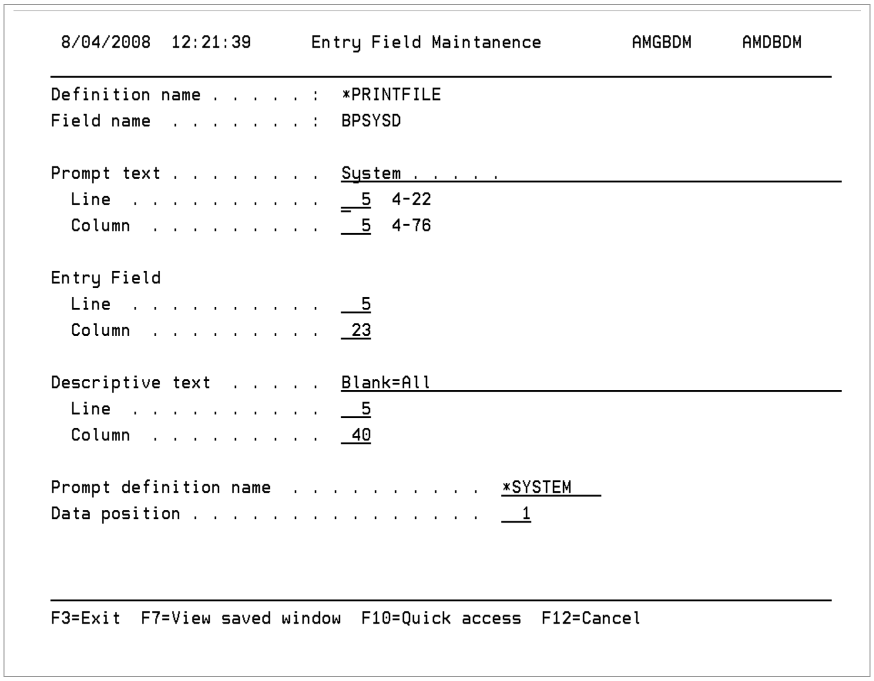
- Specifying the field specification file
- Selecting the fields you want to define for the entry panel and their type:
    - Entry field
    - Selection field



```
 8/04/2008  12:20:29      Entry Panel Definition        AMGBDM     AMDBDM
________________________________________________________________________________
Entry panel definition name  . . . . . . :   *PRINTFILE

Field specification file . . . . . . . . .   AMLBP
  Library  . . . . . . . . . . . . . . . .   *LIBL

Type options, press Enter.
  4=Delete   5=Define panel   8=Define validity checking   9=Language overrides

Opt Field       Line     Column    Prompt Text               Type      Length
 =  BPSYSD         5       23      System . . . . .          Char         2
 _  BPUSER         7       23      User . . . . . .          Char        10
 _  BPFILE         9       23      Print File . . .          Char        10
 _  BPTSYS                         To system  . . .          Char         2
 _  BPSYSV                         Version  . . . .          Packed       3,0
 _  BPTVER                         To version . . .          Packed       3,0
 _  BPJOBN                         Job name . . . .          Char        10
 _  BPTJOB                         To job name  . .          Char        10  +


________________________________________________________________________________
F3=Exit  F5=Refresh  F7=View saved window  F10=Quick access  F24=More keys
```

Figure 6-4: Entry Panel Definition screen

From this screen you identify the file and select whether to define the panel, define validity checking, or specify a language override. Refer to the "Working with Language Overrides" chapter for information on language overrides.



```
 8/04/2008  12:21:39      Entry Field Maintanence         AMGBDM     AMDBDM

Definition name . . . . . :  *PRINTFILE
Field name  . . . . . . . :  BPSYSD

Prompt text . . . . . . . .  System . . . . .
  Line  . . . . . . . . . .    5  4-22
  Column  . . . . . . . . .    5  4-76

Entry Field
  Line  . . . . . . . . . .    5
  Column  . . . . . . . . .   23

Descriptive text  . . . . .  Blank=All
  Line  . . . . . . . . . .    5
  Column  . . . . . . . . .   40

Prompt definition name  . . . . . . . . . .  *SYSTEM
Data position . . . . . . . . . . . . . . .    1



F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel
```

Figure 6-5: Entry Field Maintenance screen

You can identify the following items when you define an entry field:

- Prompt text and its location
- The entry field and its location
- Descriptive text and its location (optional)
- A prompt definition for the entry field (optional)

You can identify the following items when you define a selection field**:**

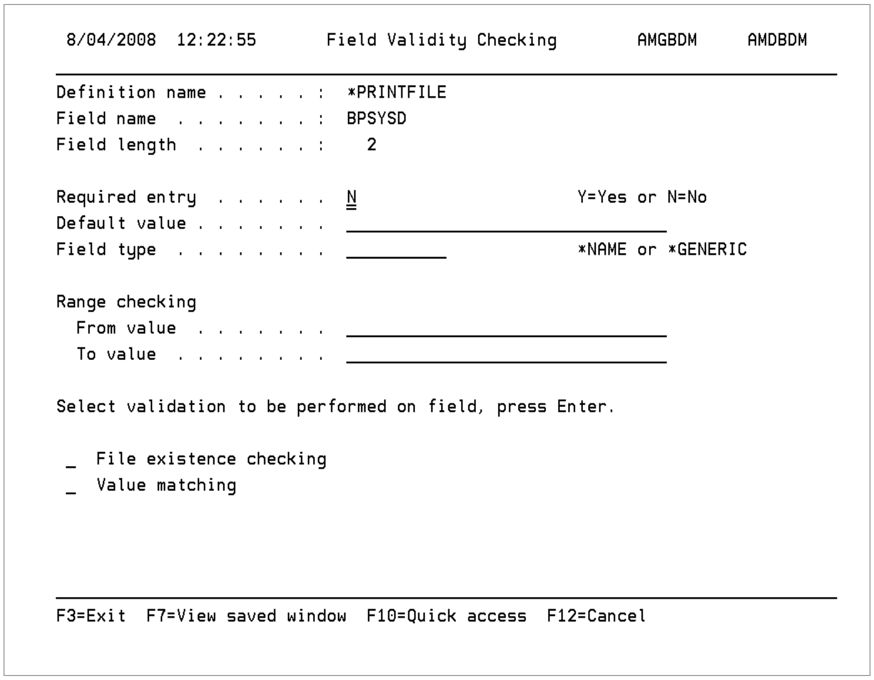
- Prompt text and its location
- The entry field and its location
- A series of selection choices

Validity checking of entry fields

You can complete the following types of validity checking on an entry field:

- Required/not required fields
- A default value
- Field types for character fields
- Range checking

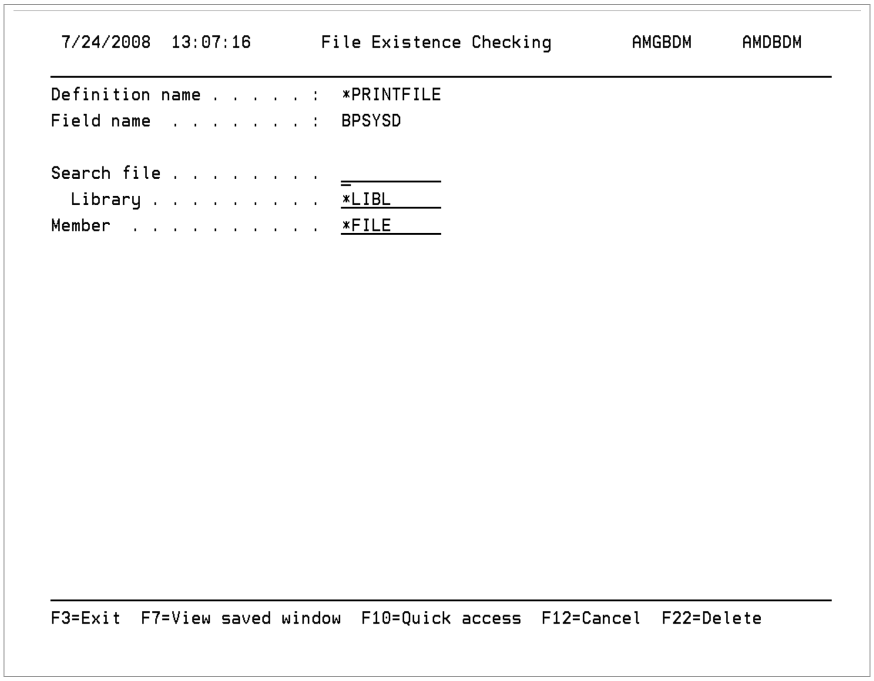
- File existence checking
- Value matching



```
 8/04/2008  12:22:55       Field Validity Checking        AMGBDM     AMDBDM
______________________________________________________________________________
Definition name . . . . . :  *PRINTFILE
Field name  . . . . . . . :  BPSYSD
Field length  . . . . . . :    2

Required entry  . . . . . .  N                      Y=Yes or N=No
Default value . . . . . . .  ________________________________
Field type  . . . . . . . .  __________             *NAME or *GENERIC

Range checking
  From value  . . . . . . .  ________________________________
  To value  . . . . . . . .  ________________________________

Select validation to be performed on field, press Enter.

 _  File existence checking
 _  Value matching



______________________________________________________________________________
F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel
```

Figure 6-6: Field Validity Checking screen

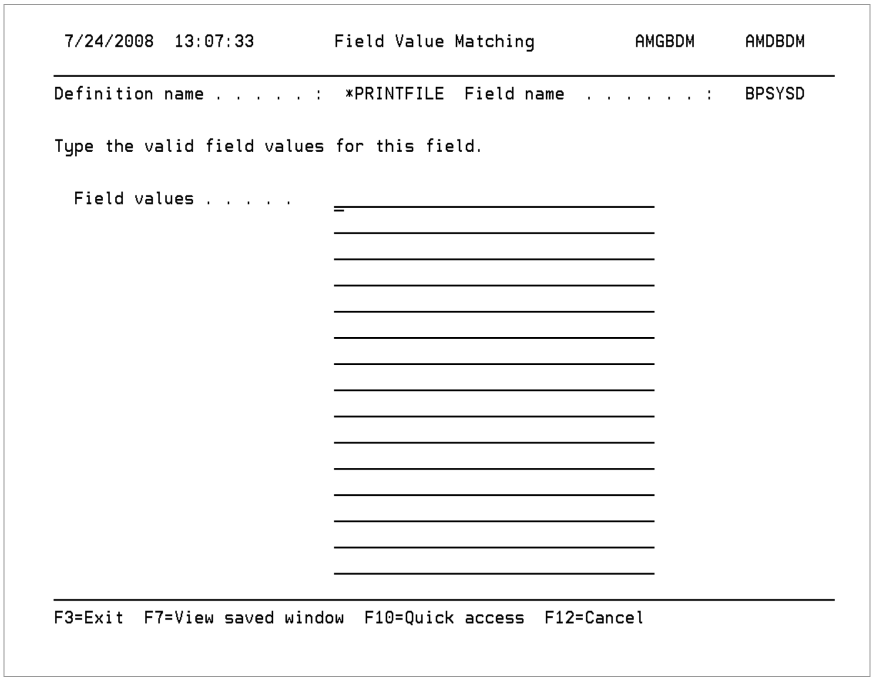
Specify basic validity checking for a field. You can also select to do file existence checking or value matching for the field.



```
 7/24/2008  13:07:16       File Existence Checking        AMGBDM     AMDBDM

Definition name . . . . . :  *PRINTFILE
Field name  . . . . . . . :  BPSYSD

Search file . . . . . . . .  __________
  Library . . . . . . . . .  *LIBL_____
Member  . . . . . . . . . .  *FILE_____












F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel  F22=Delete
```

Figure 6-7: File Existence Checking definition screen

Specify the search file that you want to check for existence of a value.



```
 7/24/2008  13:07:33       Field Value Matching         AMGBDM     AMDBDM

Definition name . . . . . :  *PRINTFILE  Field name  . . . . . . :   BPSYSD

Type the valid field values for this field.

  Field values . . . . .     ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________
                             ________________________________

F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel
```

Figure 6-8: Field Value Matching screen

You can identify the values that an entry must match to be a valid entry.

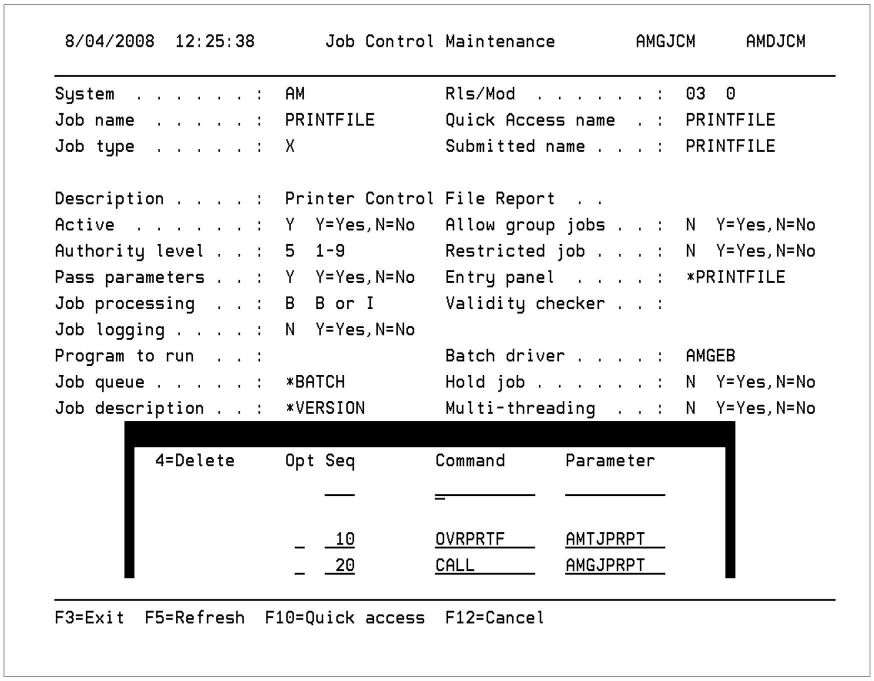
Activating entry panels

Follow these steps to activate an entry panel:

1 Create an entry panel definition.

2 Add the entry panel definition to a job control record.

3 Add code to the proper application program.



```
 8/04/2008  12:25:38       Job Control Maintenance        AMGJCM     AMDJCM

System  . . . . . . :  AM              Rls/Mod  . . . . . . :  03  0
Job name  . . . . . :  PRINTFILE       Quick Access name  . :  PRINTFILE
Job type  . . . . . :  X               Submitted name . . . :  PRINTFILE

Description . . . . :  Printer Control File Report  . .
Active  . . . . . . :  Y  Y=Yes,N=No   Allow group jobs . . :  N  Y=Yes,N=No
Authority level . . :  5  1-9          Restricted job . . . :  N  Y=Yes,N=No
Pass parameters . . :  Y  Y=Yes,N=No   Entry panel  . . . . :  *PRINTFILE
Job processing  . . :  B  B or I       Validity checker . . :
Job logging . . . . :  N  Y=Yes,N=No
Program to run  . . :                  Batch driver . . . . :  AMGEB
Job queue . . . . . :  *BATCH          Hold job . . . . . . :  N  Y=Yes,N=No
Job description . . :  *VERSION        Multi-threading  . . :  N  Y=Yes,N=No

         4=Delete      Opt Seq         Command       Parameter
                           ___         _________     __________

                        _   10         OVRPRTF___    AMTJPRPT__
                        _   20         CALL______    AMGJPRPT__

F3=Exit  F5=Refresh  F10=Quick access  F12=Cancel
```

Figure 6-9: Job Control Maintenance screen

You need to do the following to activate an entry panel through a job control:

- Set the *Pass parameters* field to **Y** on the job control definition of the program that requires the parameters.
- Type the entry panel definition name in the *Entry panel* field.
- Specify a program that will be doing the validity checking if you are not using the validity checking within entry panels.

Validity checking specifications

The parameter passed to the validity checking program is a single 256 byte character field with the following structure:

28 byte key	28 characters
Message file	10 characters
Message ID	7 characters
Message text	80 characters
Reserved	131 characters

Infinium AM performs validity checks defined on the entry panel first and then calls the specified validity checking program.

The validity checker uses the 28 byte key to chain out to the task coupling file to retrieve the record to validate. For any error found, the validity checking program returns the message ID, message file, and any message data to the entry panel processing program.

Note: The error message sent should specify exactly what is in error since a field on the entry panel cannot be flagged.

Coding for entry panels - application program

Coding for entry panels in the application program should include the following:

1 Declare the field specification file, ADLEP.

```
 2200      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 2300     FADLFIL1 IF  E           K        DISK
 2400     FADLEP   IF  E           K        DISK
 2500     FADTWRPT O   E                    PRINTER      KINFDS PRT1
 2600     F                                              KINFSR OA
 2700      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

2 Accept the 28 character task coupling record key. The Event Manager/Requester passes this key to the application program.

```
 4800      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 4900     C           *ENTRY    PLIST
 5000     C                     PARM           PARM28 28
 5100      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 5200      *   MAIN PROGRAM
```

3 Code how you will use the passed parameters by using the field names from the field specification file, ADPEP.

```
 5700      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 5800      *   MAIN PROCESS
 5900      *
 6000     C           PROCES    BEGSR
 6100     C                     EXSR OA
 6200     C                     READ ADLFIL1                  99
 6300     C           *IN99     DOWEQ'0'
 6400     C                     EXSR RANGE
 6500     C           OUTSID    IFEQ '0'
 6600     C           STLINE    CASGEFO1       OA
 6700     C           *IN51     CASEQ'1'       OA
 6800     C                     ENDCS
 6900     C                     WRITEADRWDET                51
 7000     C                     ENDIF
 7100     C                     READ ADLFIL1                  99
 7200     C                     ENDDO
 7300      *
 7400     C                     ENDSR
 7500      *
```

4 Read the field specification file, ADLEP, for values of the input fields

```
12500      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
12600      *INITIALIZATION
12700      *
```

```
12800     C           XINIT     BEGSR
12900      *
13000     C                     Z-ADD0         FO1     30
13100     C                     Z-ADD0         FO7     30
13200     C                     MOVE '0'       OUTSID  1
13300     C                     MOVE 1         *IN51
13700     C           PARM28    CHAINADLEP                96
13800     C           *IN96     IFEQ '1'                        NOT FOUND
13900     C                     RETRN
14000     C                     END
```

Chapter 7 Prompts

7

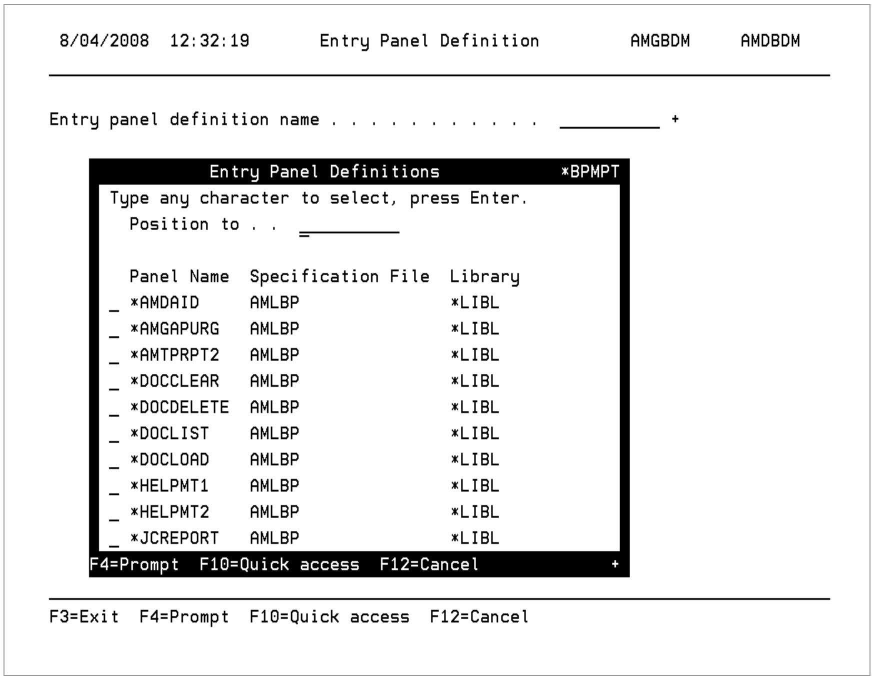
When you complete this chapter, you will know how to define a prompt with Infinium AM and activate a prompt by modifying the application program code.

The chapter consists of the following topics:

Overview

A prompt definition of a window can be:

- Displayed from a field or menu
- Displayed as a full screen window or a partial screen window
- Used to return a value to the screen



```
 8/04/2008  12:32:19       Entry Panel Definition         AMGBDM     AMDBDM

Entry panel definition name . . . . . . . . . . .  __________ +

    Entry Panel Definitions          *BPMPT
    Type any character to select, press Enter.
      Position to . .  __________

      Panel Name  Specification File  Library
    _ *AMDAID     AMLBP               *LIBL
    _ *AMGAPURG   AMLBP               *LIBL
    _ *AMTPRPT2   AMLBP               *LIBL
    _ *DOCCLEAR   AMLBP               *LIBL
    _ *DOCDELETE  AMLBP               *LIBL
    _ *DOCLIST    AMLBP               *LIBL
    _ *DOCLOAD    AMLBP               *LIBL
    _ *HELPMT1    AMLBP               *LIBL
    _ *HELPMT2    AMLBP               *LIBL
    _ *JCREPORT   AMLBP               *LIBL
    F4=Prompt  F10=Quick access  F12=Cancel         +

F3=Exit  F4=Prompt  F10=Quick access  F12=Cancel
```

Figure 7-1: Example of a prompt screen

When the user presses F4, the prompt screen or window is displayed. By entering **X** the user can return the value to the screen*.*

Assigning a definition

The type of prompt windows are:

- An inquiry prompt that only displays fields
- An input prompt that has selection fields and provides for the return of window data

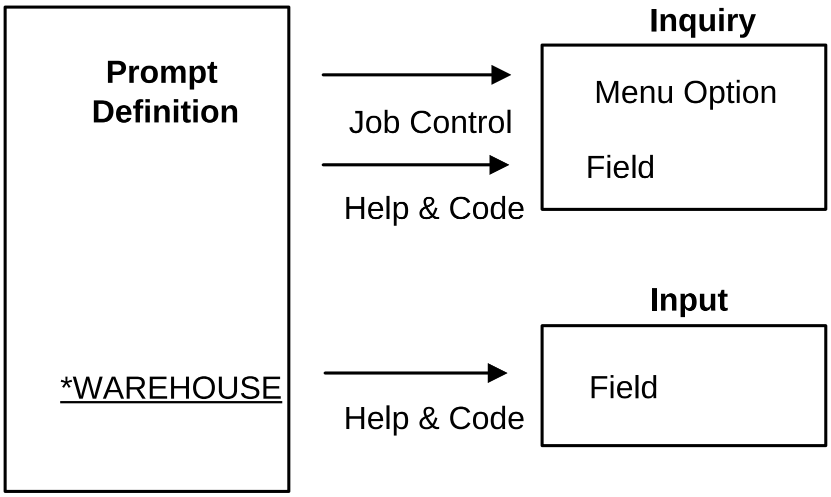


Figure 7-2: Types of prompt windows

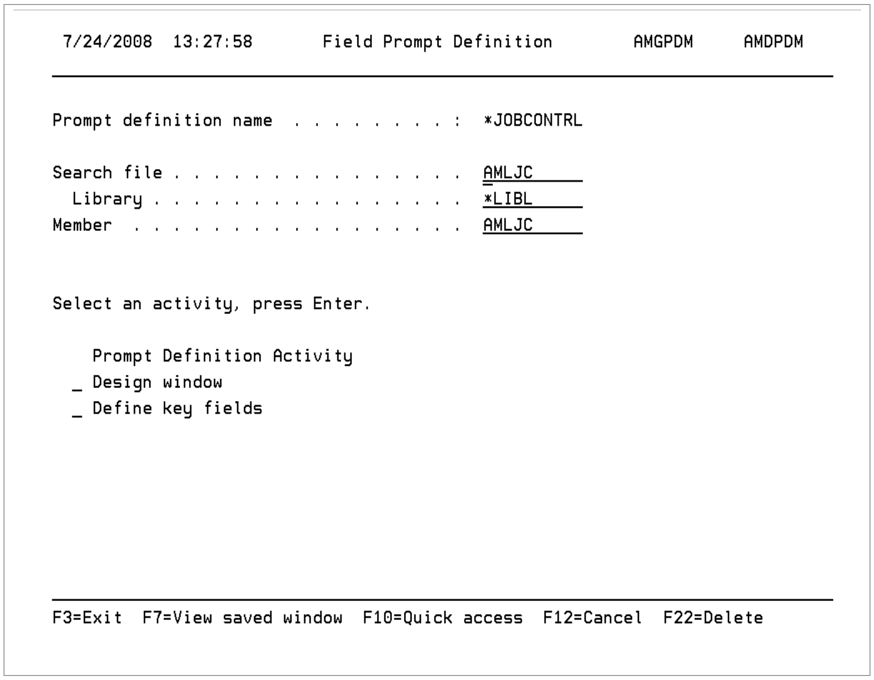
You can assign a prompt definition in two ways:

- To a menu through a job control
- To a field through the help system and by adding code to the application program

Creating a prompt definition

There components to a prompt definition are:

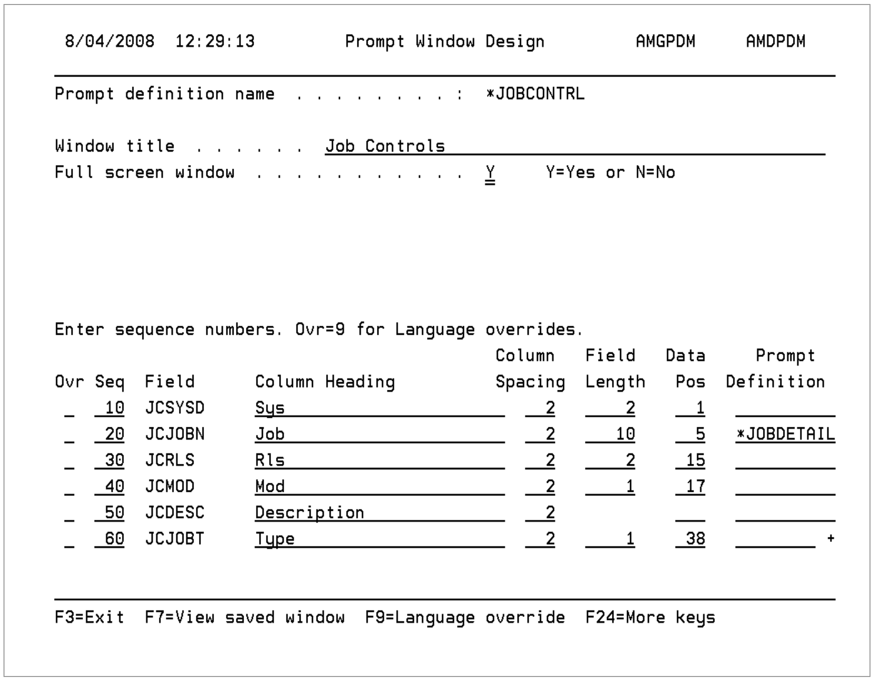
- Specifying the prompt definition names and search file
- Designing a window
- Defining key fields



```
 7/24/2008  13:27:58       Field Prompt Definition        AMGPDM     AMDPDM

Prompt definition name  . . . . . . . . :  *JOBCONTRL

Search file . . . . . . . . . . . . . . .  AMLJC
  Library . . . . . . . . . . . . . . . .  *LIBL
Member  . . . . . . . . . . . . . . . . .  AMLJC


Select an activity, press Enter.

    Prompt Definition Activity
  _ Design window
  _ Define key fields








F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel  F22=Delete
```

Figure 7-3: Field Prompt Definition screen

From this screen you identify the file and select whether to design a window or define key fields.



```
 8/04/2008  12:29:13         Prompt Window Design        AMGPDM     AMDPDM
______________________________________________________________________________
Prompt definition name  . . . . . . . . . :  *JOBCONTRL

Window title  . . . . . . .  Job Controls_____________________________________
Full screen window  . . . . . . . . . . . .  Y      Y=Yes or N=No





Enter sequence numbers. Ovr=9 for Language overrides.
                                            Column   Field   Data    Prompt
Ovr Seq  Field      Column Heading          Spacing  Length   Pos  Definition
 _   10  JCSYSD     Sys_______________________  2       2      1   __________
 _   20  JCJOBN     Job_______________________  2      10      5   *JOBDETAIL
 _   30  JCRLS      Rls_______________________  2       2     15   __________
 _   40  JCMOD      Mod_______________________  2       1     17   __________
 _   50  JCDESC     Description_______________  2             __   __________
 _   60  JCJOBT     Type______________________  2       1     38   ________  +

______________________________________________________________________________
F3=Exit  F7=View saved window  F9=Language override  F24=More keys
```

Figure 7-4: Prompt Window Design screen

When you design a window, you can identify the following:

- Which fields to display from the search file by typing the sequence number (do not select an encrypted field such as a password)
- The order of the fields to display as columns in the window
- The text of the column heading
- The column/field length and column spacing
- The size of the window
- The position of the field value in an input/return data structure
- Whether the window has a nested prompt

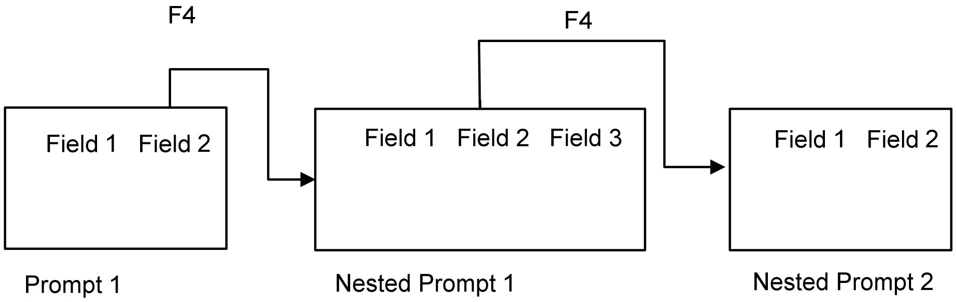
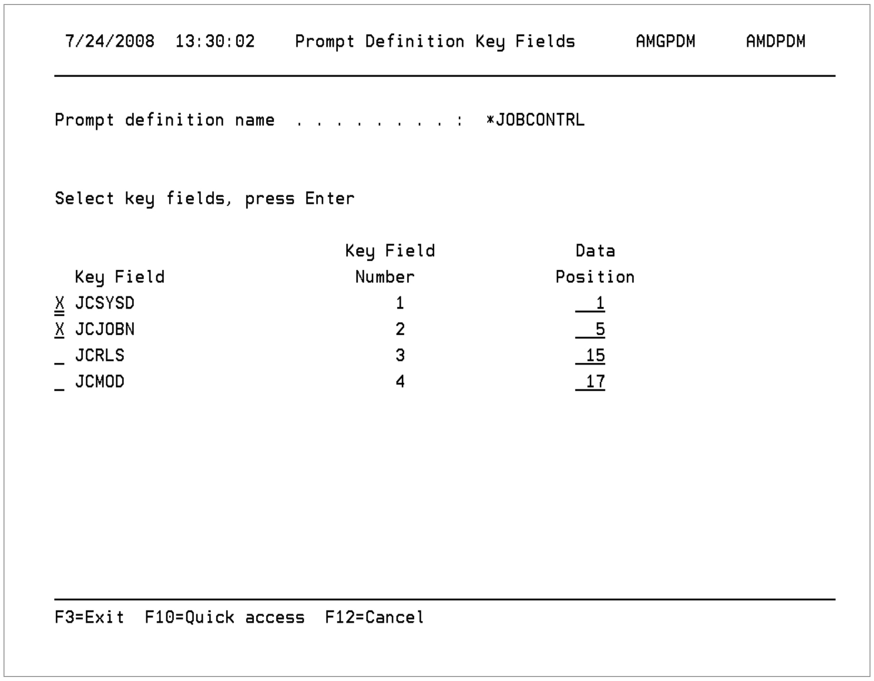


Figure 7-5: Example of nested prompts



```
 7/24/2008  13:30:02    Prompt Definition Key Fields      AMGPDM     AMDPDM

Prompt definition name  . . . . . . . . :  *JOBCONTRL


Select key fields, press Enter

                             Key Field            Data
  Key Field                   Number            Position
X JCSYSD                        1                   1
X JCJOBN                        2                   5
_ JCRLS                         3                  15
_ JCMOD                         4                  17








F3=Exit  F10=Quick access  F12=Cancel
```

Figure 7-6: Prompt Definition Key Fields screen

You need to define key fields only if you typed a logical file for the search file.

To define key fields, use the following information:

- The system displays a list of key fields from within the logical view.
- Select a subset of the key fields for a partial key lookup.
- You must specify the key fields in sequence.
- Data position is required in order to locate the key field value in the input portion of the data structure passed to the run time prompt software.

Activating menu based prompts

Follow these steps to activate a menu based prompt:

1. Create a prompt definition.
2. Create a job control definition.
3. Add the job control definition to the appropriate system, version, or user menu.

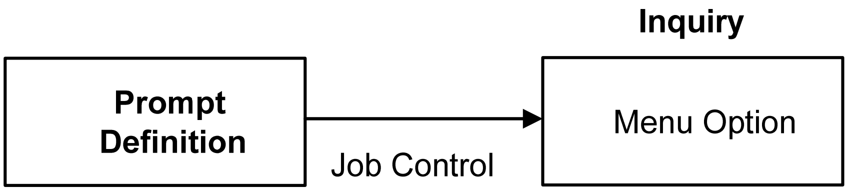
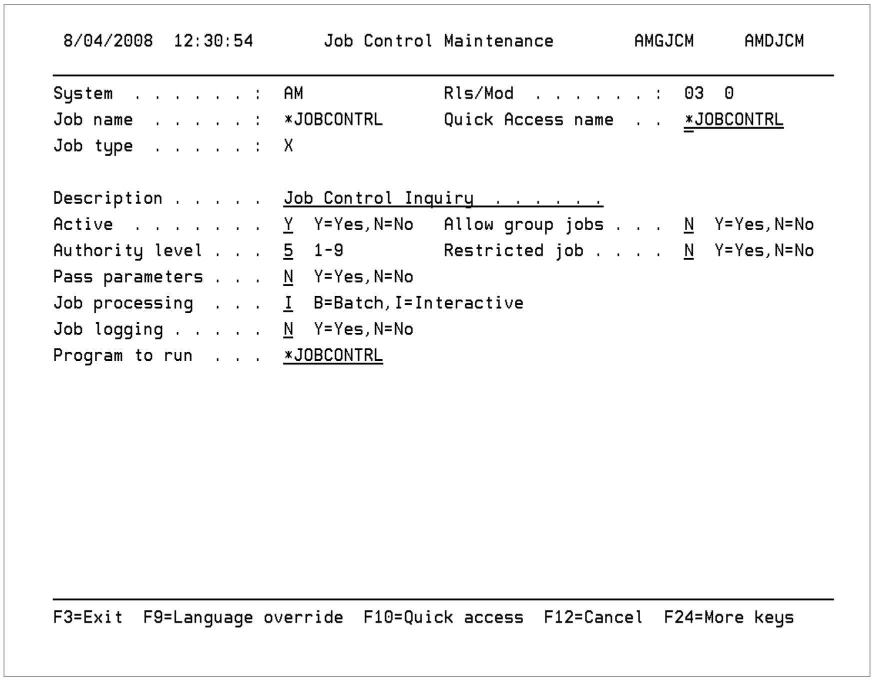


Figure 7-7: Activating a menu based prompt



```
 8/04/2008  12:30:54       Job Control Maintenance        AMGJCM     AMDJCM

System  . . . . . . :  AM              Rls/Mod  . . . . . . :  03  0
Job name  . . . . . :  *JOBCONTRL      Quick Access name  . .  *JOBCONTRL
Job type  . . . . . :  X

Description . . . . .  Job Control Inquiry  . . . . . .
Active  . . . . . . .  Y  Y=Yes,N=No   Allow group jobs . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9          Restricted job . . . .  N  Y=Yes,N=No
Pass parameters . . .  N  Y=Yes,N=No
Job processing  . . .  I  B=Batch,I=Interactive
Job logging . . . . .  N  Y=Yes,N=No
Program to run  . . .  *JOBCONTRL

F3=Exit  F9=Language override  F10=Quick access  F12=Cancel  F24=More keys
```

Figure 7-8: Job Control Maintenance screen

Specify an interactive job and type the prompt definition name in the *Program to run* field.

Activating field based prompts

Follow these steps to activate a field based prompt:

1. Create a prompt definition.
2. Assign the prompt definition to the field's help text.
3. Add code to the proper application program as follows:
   - Inquiry prompt

     Add code to handle the prompt function key.
   - Input prompt

     Add the above code as well as code to set the input portion of the prompt data structure and to retrieve data from the return portion of the prompt data structure.

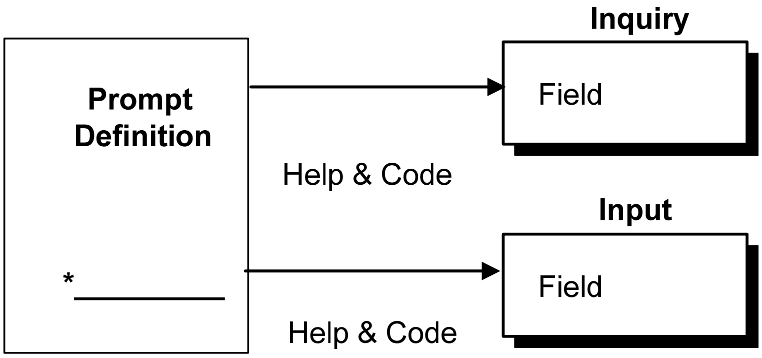
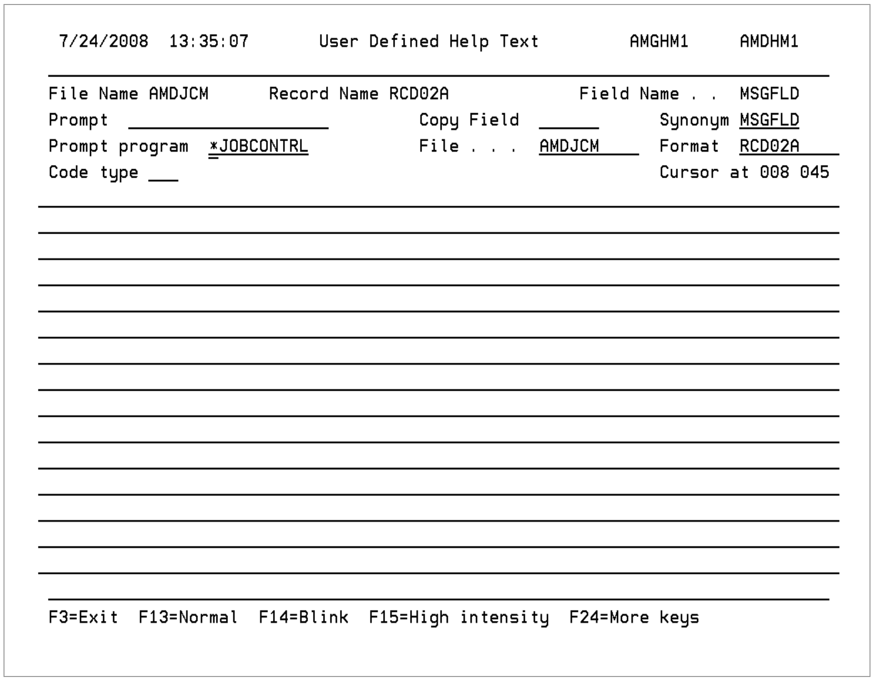


Figure 7-9: Activating field based prompts



```
 7/24/2008  13:35:07       User Defined Help Text         AMGHM1     AMDHM1

File Name AMDJCM      Record Name RCD02A             Field Name . .   MSGFLD
Prompt                                 Copy Field             Synonym MSGFLD
Prompt program  *JOBCONTRL             File . . .  AMDJCM     Format  RCD02A
Code type                                                     Cursor at 008 045

F3=Exit  F13=Normal  F14=Blink  F15=High intensity  F24=More keys
```

Figure 7-10: User Defined Help Text screen

Position the cursor on the field in your system where to add a field based prompt and complete the following:

- Press Help.
- Press F6.
- Type the prompt definition name in the *Prompt program* field.

Note: You must have authority to update help text.

Coding for inquiry prompts

Add the following line of code to handle an inquiry prompt:

$ACTN   CASEQ 'PROMPT' $PRMPT

It is a good practice to add this line of code to each screen in your application.

Sample screen procedure

```
C                     EXSR $FT2
 *1
 *1 The following hard coded key actions
 *1 need to be changed to use Soft Coder.
 *1
C           $ACTN     CASEQ'EXIT'    EXIT
C           $ACTN     CASEQ'DSPVND'  DSPVND
C           $ACTN     CASEQ'RETURN'  EXIT
C           $ACTN     CASEQ'ENTER'   PROC02
C           $ACTN     CASEQ'*PROMPT' PRMPT
C           $ACTN     CASEQ'*HELP'   $ALIGN
C           $ACTN     CASEQ'*MORKYS' MORKYS
C                     ENDCS
```

$PRMPT Subroutine (from $AMSTD)

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C           $PRMPT    BEGSR
 *
C           STCSRP    DIV  256       $ROW
C                     MVR            $COL
 *
C                     CALL 'AMGPI1'              69
C                     PARM           STDFIL
C                     PARM           STDFMT
C                     PARM           $ROW    30
C                     PARM           $COL    30
C                     PARM           $TYPE   1
C                     PARM           $DATA 256
C                     PARM           $FLDN   6
 *
C                     MOVE '1'       *IN48
 *
C                     ENDSR
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Coding for input prompts

You need to complete the following to handle an input prompt:

- Create an input and return data structure.
- Add the input and return fields to the field reference file.
- Add the necessary code to the RPG program.

Input/return data structure

All programs that utilize the input/return capabilities of prompt must pass and receive a 256-byte data structure set up in the following way:

- The low 128 is input.
- The high 128 is return.

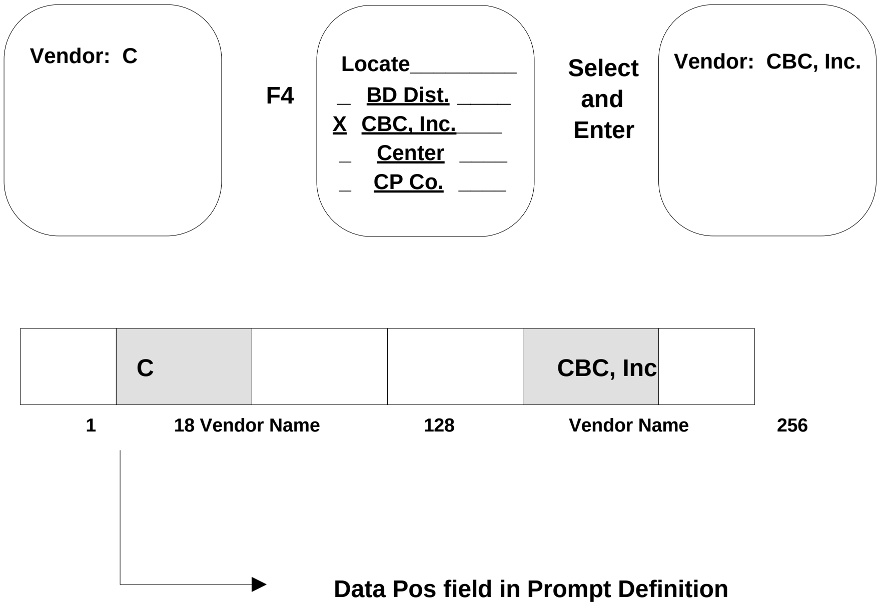


Figure 7-11: Input/Return Data Structure

Creating the input and return data structure

As a one time setup, you need to create the input and return data structure. You can do this as follows:

- Select the fields in the application you will want to prompt.
- Create a file for the input of each field and a file for the return of each field. Using @I and @R before the field names is a suggested naming convention.
- Type the input and return fields in the field specification file.

Sample input data structure file - ADSPI

```
A                                      REF(ADFLDREF)
A          R DPARMS
 *
 *  INPUT FIELDS FOR 128 DATA STRUCTURE...
 *
A            @ICUSR    R
A            @IVEND    R
A            @IWRHS    R
A            @ILOC     R
A            @IPART    R
A            @IDESC    R
A            @INULL    R
```

Sample return data structure file - ADSPR

```
A                                      REF(ADFLDREF
A          R DPARMS
 *
 *  RETURN FIELDS FOR 128 DATA STRUCTURE...
 *
A            @RCUSR    R
A            @RVEND    R
A            @RWRHS    R
A            @RLOC     R
A            @RPART    R
A            @RDESC    R
A            @RNULL    R
```

Addition to field reference file

```
 *
A            @ICUSR        10
A            @IVEND        10
A            @IWRHS         3
A            @ILOC          2
A            @IPART         6
```

```
6800     A            @IDESC        20
6900      *
7900      *
8000     A            @RCUSR        10
8100     A            @RVEND        10
8200     A            @RWRHS         3
8300     A            @RLOC          2
8400     A            @RPART         6
8500     A            @RDESC        20
8600      *
```

Coding in the program

There are three things to accomplish in the RPG program:

- Declare the input and return data structure
- Handle the prompt function key
- Retrieve data from the return prompt data structure

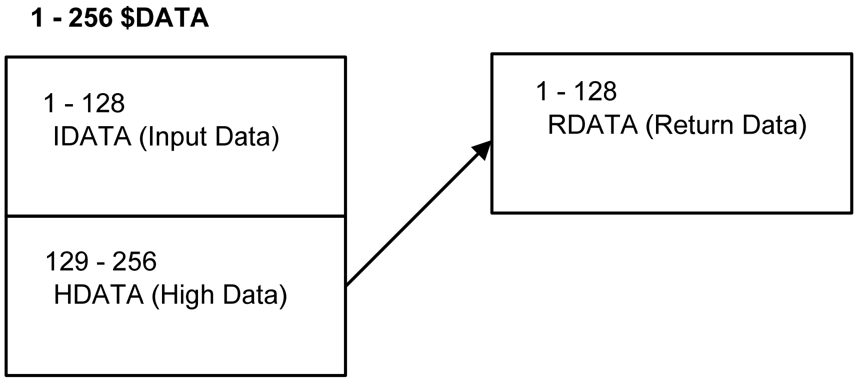


Figure 7-12: Relationship between input and return data structures

Declare input and return data structure

```
5400     I          E DSADSPI
5500     I                                 1 256 $DATA
5600     I                               129 256 HDATA
5700     I          E DSADSPR
5800     I                                 1 128 RDATA
5900      *
```

Retrieving data

You can structure the PRMPT subroutine to handle input and return data (field values) for:

- A single field
- Multiple fields

Sample PRMPT subroutine multiple field

```
32900     C           PRMPT     BEGSR
33000      *
33100     C                     MOVE *BLANK    $DATA
33200     C                     MOVE FLWRHS    @IWRHS
33300      *
33400     C                     EXSR $PRMPT
33500     C                     MOVE HDATA     RDATA
33600      *
33700     C           @PL,P     IFEQ 'PROC01'
33800     C           @RWRHS    IFNE *
33900     C                     MOVE @RWRHS    R1WRHS
34000     C                     ENDIF
34100     C           @RLOC     IFNE *BLANK
34200     C                     MOVE @RLOC     R1LOC
34300     C                     ENDIF
34400     C           @RPART    IFNE *BLANK
34500     C                     MOVE @RPART    R1PART
34600     C                     ENDIF
34700     C           @RVEND    IFNE *BLANK
34800     C                     MOVE @RVEND    R1VEND
34900     C                     ENDIF
35000     C                     ENDIF
35100      *
35200     C                     ENDSR
```

Chapter 8 Working With Language Overrides

8

The chapter consists of the following topics:

Overview

All Infinium applications use a single code base for working with multiple languages.

All of the literals were extracted from the Infinium application source code and database files. Infinium supplies these literals entirely via message files, which are resolved at execution time.

Infinium delivers all of the message files in English. Languages other than English are licensed and installed separately. Refer to the *Installing a New Infinium Application* and the *Upgrading an Infinium Application* guides for information on working with alternate language files.

Note: Infinium AM does not support the entry of double-byte data.

Maintaining language overrides

Infinium applications use message ID processing in the following functions:

- Entry panels
- Function keys
- Job controls
- Menus
- News
- Prompts

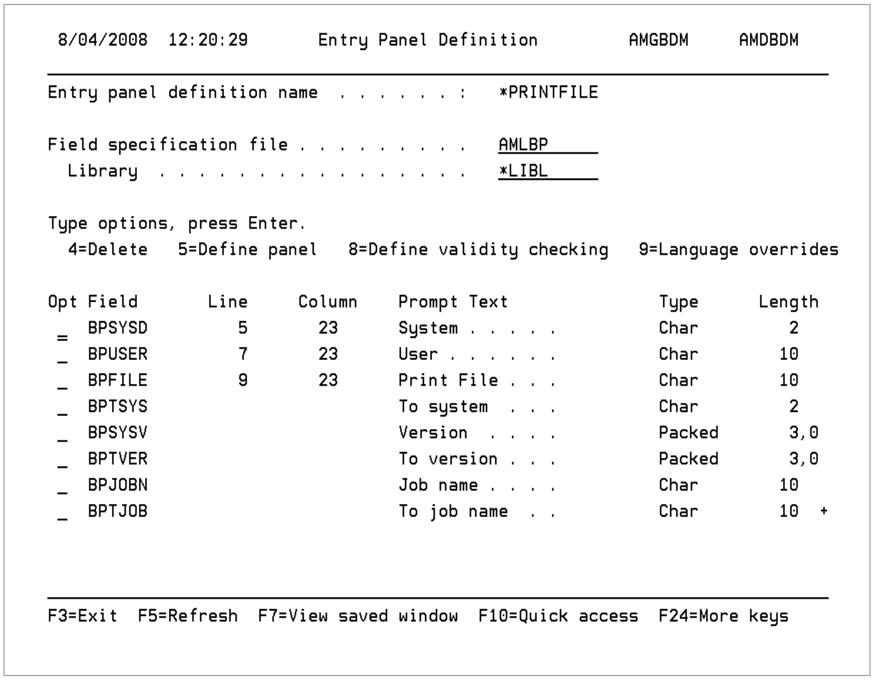
Complete the tasks in this section if you need to create and maintain language overrides.

Working with entry panel language overrides

Complete the steps below to manually add entry panel language overrides.

1 From the Infinium AM main menu, select *Entry Panels*.

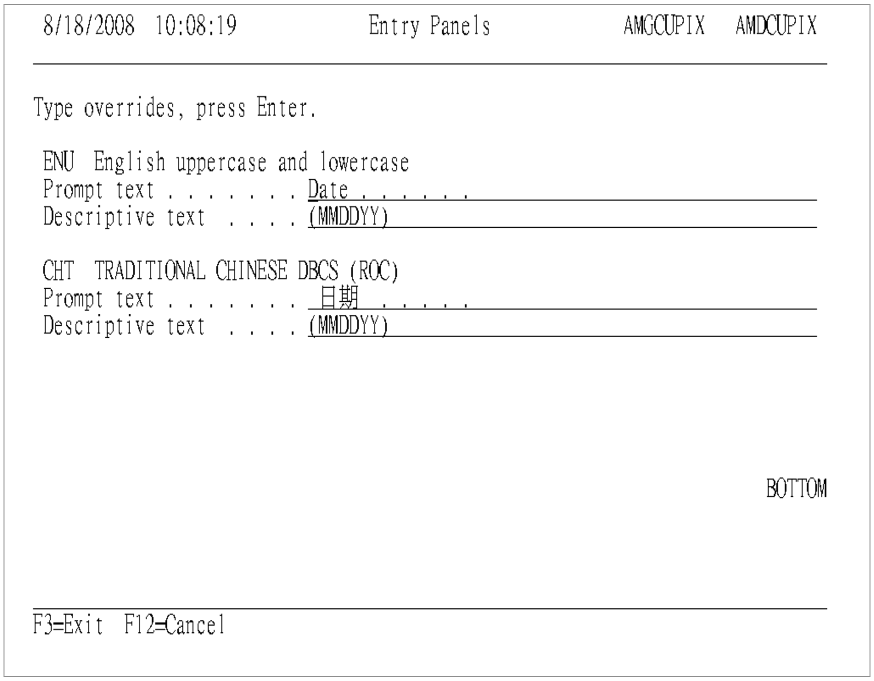
2 Specify an entry panel and press Enter. The system displays a screen similar to Figure 8-1.



```
 8/04/2008  12:20:29       Entry Panel Definition         AMGBDM     AMDBDM
______________________________________________________________________________
Entry panel definition name  . . . . . . . :   *PRINTFILE

Field specification file . . . . . . . . .   AMLBP_____
  Library  . . . . . . . . . . . . . . . .   *LIBL_____

Type options, press Enter.
  4=Delete   5=Define panel   8=Define validity checking   9=Language overrides

Opt Field       Line     Column     Prompt Text              Type      Length
 =  BPSYSD        5        23       System . . . . .         Char         2
 _  BPUSER        7        23       User . . . . . .         Char        10
 _  BPFILE        9        23       Print File . . .         Char        10
 _  BPTSYS                          To system  . . .         Char         2
 _  BPSYSV                          Version  . . . .         Packed       3,0
 _  BPTVER                          To version . . .         Packed       3,0
 _  BPJOBN                          Job name . . . .         Char        10
 _  BPTJOB                          To job name  . .         Char        10  +


______________________________________________________________________________
F3=Exit  F5=Refresh  F7=View saved window  F10=Quick access  F24=More keys
```

Figure 8-1: Entry Panel Definition screen

3 Type **9** next to a field to maintain the message text and press Enter. The system displays a screen similar to Figure 8-2.



```
8/18/2008  10:08:19           Entry Panels           AMGCUPIX   AMDCUPIX

Type overrides, press Enter.

 ENU  English uppercase and lowercase
 Prompt text . . . . . . . Date . . . . . .
 Descriptive text  . . . . (MMDDYY)

 CHT  TRADITIONAL CHINESE DBCS (ROC)
 Prompt text . . . . . . .  日期  . . . . .
 Descriptive text  . . . . (MMDDYY)

                                                                  BOTTOM

F3=Exit  F12=Cancel
```

Figure 8-2: Entry Panels language override screen

4 Use the information below to complete the fields on this screen.

Prompt text

To override the current prompt text, type the new text.

Descriptive text

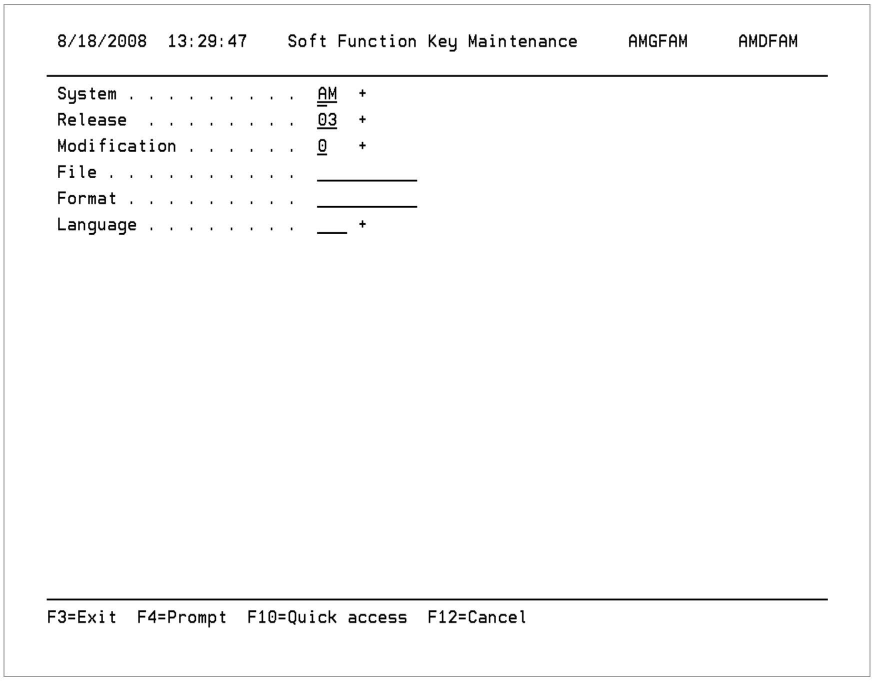
To override the description of the current prompt text, type a new description.

5 Press Enter.

Working with function key language overrides

Complete the steps below to manually add function key language overrides.

1 From the Infinium AM main menu, select *Function Keys*. The system displays a screen similar to Figure 8-3.

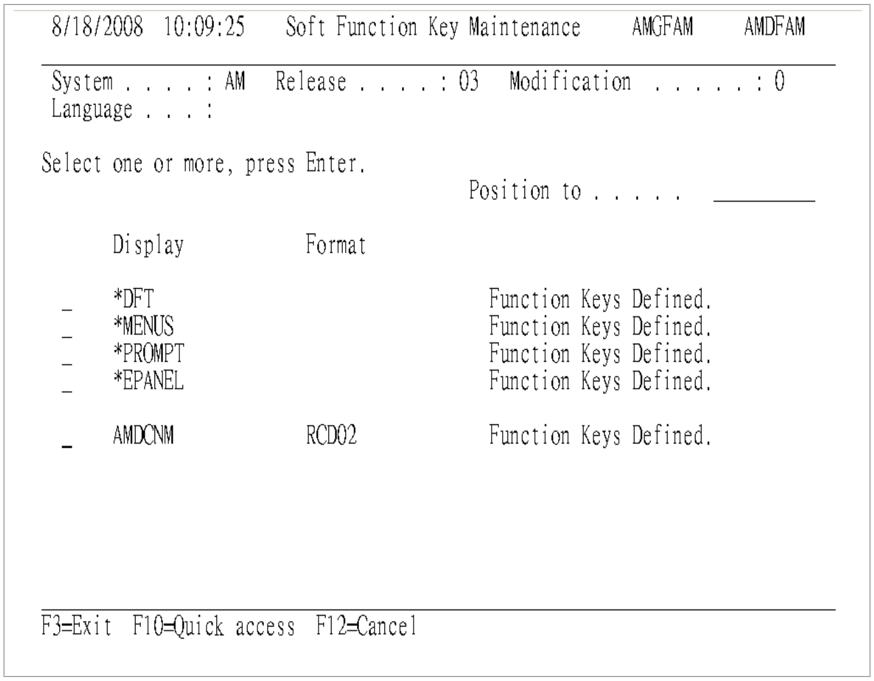


```
 8/18/2008  13:29:47    Soft Function Key Maintenance     AMGFAM     AMDFAM
____________________________________________________________________________
 System . . . . . . . . .  AM  +
 Release  . . . . . . . .  03  +
 Modification . . . . . .  0   +
 File . . . . . . . . . .  __________
 Format . . . . . . . . .  __________
 Language . . . . . . . .  ___ +















____________________________________________________________________________
F3=Exit  F4=Prompt  F10=Quick access  F12=Cancel
```

Figure 8-3: Soft Function Key Maintenance screen

2 Specify a system, release, and modification. Do not specify a language. Message IDs are language-independent.

3 Press Enter. The system displays a screen similar to Figure 8-4.

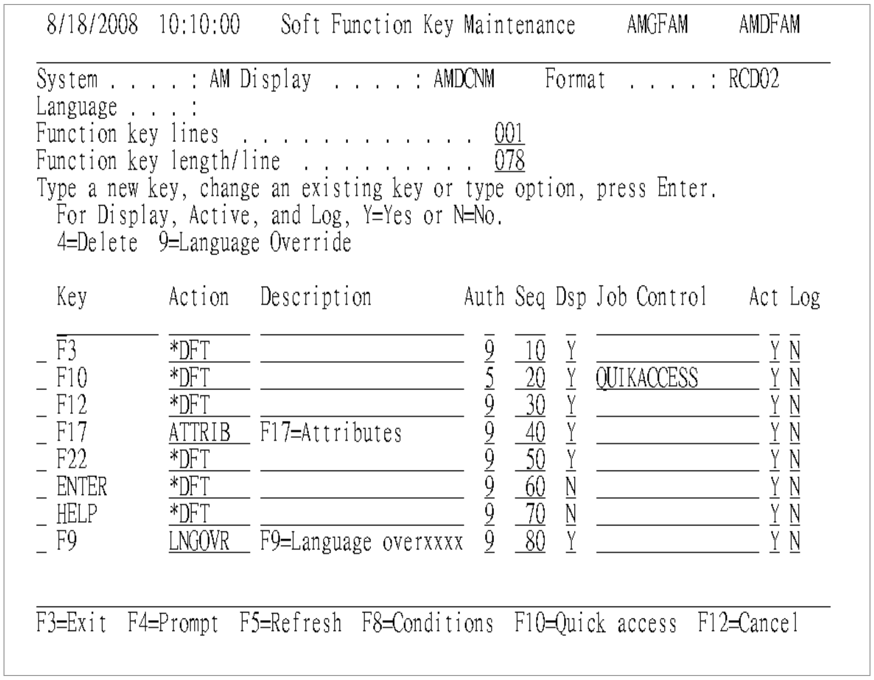


```
 8/18/2008  10:09:25    Soft Function Key Maintenance     AMGFAM     AMDFAM

System . . . . : AM   Release . . . . : 03   Modification  . . . . . : 0
Language . . . :

Select one or more, press Enter.
                                          Position to . . . . .   _________

      Display          Format

 _    *DFT                                  Function Keys Defined.
 _    *MENUS                                Function Keys Defined.
 _    *PROMPT                               Function Keys Defined.
 _    *EPANEL                               Function Keys Defined.

 _    AMDCNM           RCD02                Function Keys Defined.






F3=Exit  F10=Quick access  F12=Cancel
```

Figure 8-4: Soft Function Key Maintenance screen

4 Select a display file/record format for defining function keys.

5 Press Enter. The system displays a screen similar to Figure 8-5.

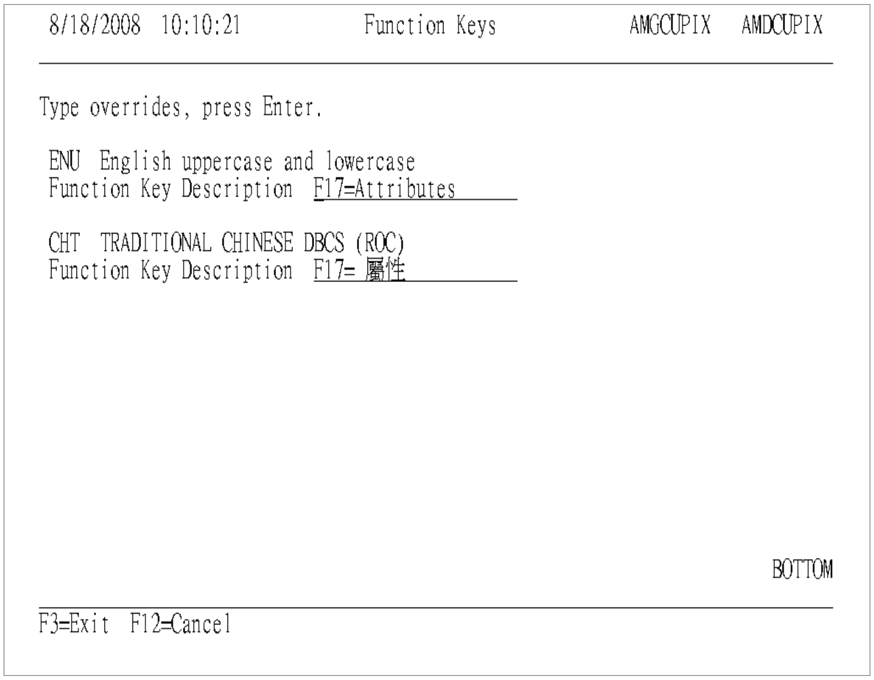


```
 8/18/2008  10:10:00    Soft Function Key Maintenance     AMGFAM     AMDFAM

System . . . . : AM Display  . . . . : AMDCNM    Format  . . . . : RCD02
Language . . . :
Function key lines  . . . . . . . . . . . .  001
Function key length/line  . . . . . . . . .  078
Type a new key, change an existing key or type option, press Enter.
  For Display, Active, and Log, Y=Yes or N=No.
  4=Delete  9=Language Override

  Key        Action   Description         Auth Seq Dsp Job Control     Act Log

_ F3         *DFT                          9   10   Y                   Y  N
_ F10        *DFT                          5   20   Y  QUIKACCESS       Y  N
_ F12        *DFT                          9   30   Y                   Y  N
_ F17        ATTRIB   F17=Attributes       9   40   Y                   Y  N
_ F22        *DFT                          9   50   Y                   Y  N
_ ENTER      *DFT                          9   60   N                   Y  N
_ HELP       *DFT                          9   70   N                   Y  N
_ F9         LNGOVR   F9=Language overxxxx 9   80   Y                   Y  N


F3=Exit  F4=Prompt  F5=Refresh  F8=Conditions  F10=Quick access  F12=Cancel
```

Figure 8-5: Soft Function Key Maintenance screen

6 Type **9** next to the record for which to maintain the message text.

7 Press Enter. The system displays a screen similar to Figure 8-6.



```
8/18/2008  10:10:21            Function Keys            AMGCUPIX   AMDCUPIX

Type overrides, press Enter.

 ENU  English uppercase and lowercase
 Function Key Description  F17=Attributes

 CHT  TRADITIONAL CHINESE DBCS (ROC)
 Function Key Description  F17= 屬性

                                                                     BOTTOM
F3=Exit  F12=Cancel
```

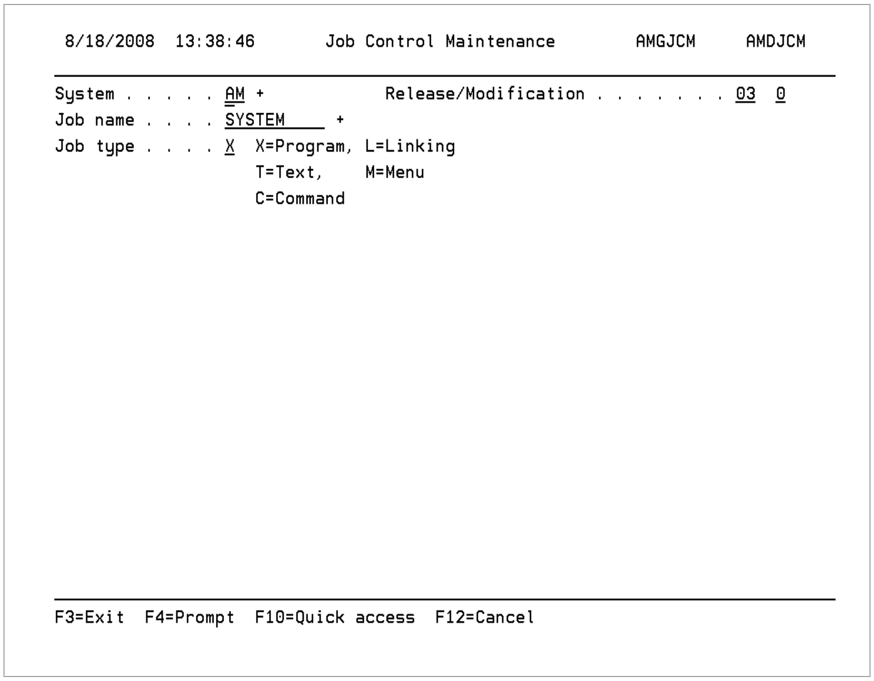
Figure 8-6: Function Keys language override screen

8 To override the function key description, type the new text and press Enter.

Working with job control language overrides

Complete the steps below to manually add job control language overrides.

1 From the Infinium AM main menu, select *Job Controls*. The system displays a screen similar to Figure 8-7.

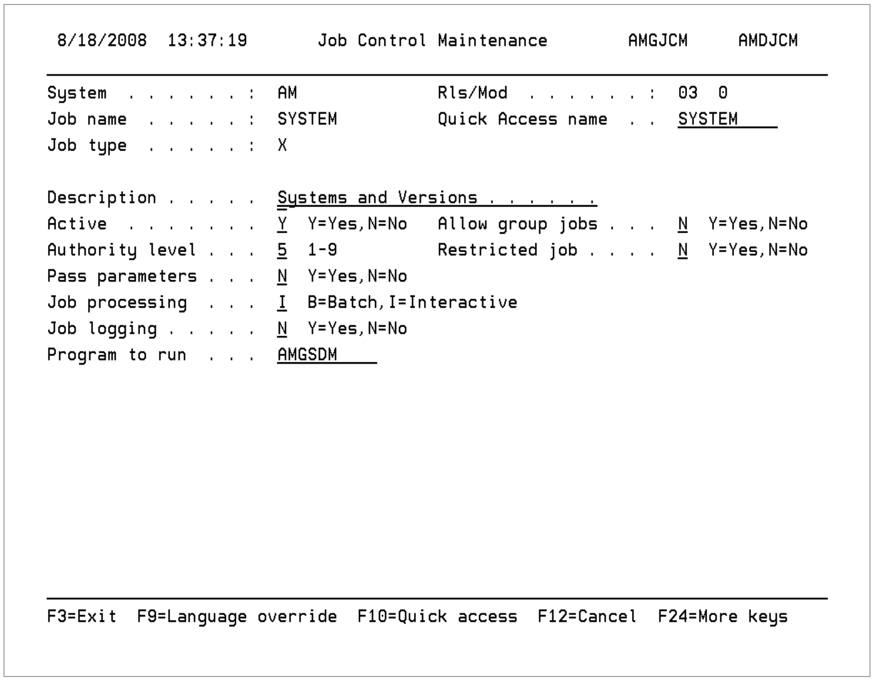


```
 8/18/2008  13:38:46       Job Control Maintenance        AMGJCM     AMDJCM

System . . . . . AM +           Release/Modification . . . . . . . 03  0
Job name . . . . SYSTEM     +
Job type . . . . X  X=Program, L=Linking
                    T=Text,    M=Menu
                    C=Command

F3=Exit  F4=Prompt  F10=Quick access  F12=Cancel
```

Figure 8-7: Job Control Maintenance screen

2 Specify a system, release, modification, job name, and type.

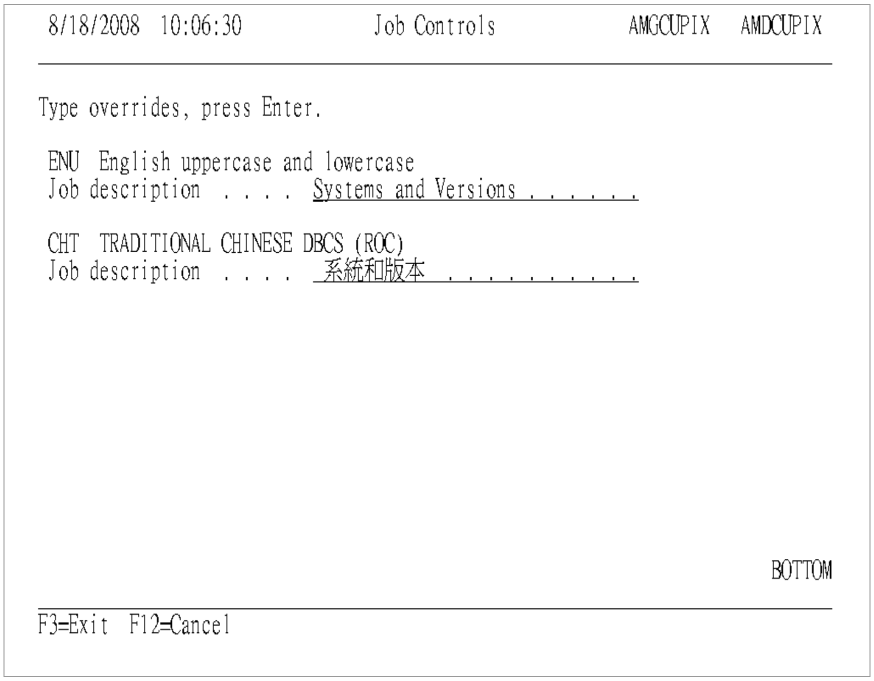
3 Press Enter. The system displays a screen similar to Figure 8-8.



```
 8/18/2008  13:37:19      Job Control Maintenance       AMGJCM     AMDJCM

System  . . . . . . :  AM               Rls/Mod  . . . . . . :  03  0
Job name  . . . . . :  SYSTEM           Quick Access name  . .  SYSTEM
Job type  . . . . . :  X

Description . . . . .  Systems and Versions . . . . . .
Active  . . . . . . .  Y  Y=Yes,N=No    Allow group jobs . . .  N  Y=Yes,N=No
Authority level . . .  5  1-9           Restricted job . . . .  N  Y=Yes,N=No
Pass parameters . . .  N  Y=Yes,N=No
Job processing  . . .  I  B=Batch,I=Interactive
Job logging . . . . .  N  Y=Yes,N=No
Program to run  . . .  AMGSDM


F3=Exit  F9=Language override  F10=Quick access  F12=Cancel  F24=More keys
```

Figure 8-8: Job Control Maintenance screen

4 Press F9 to maintain the message text.



```
 8/18/2008  10:06:30            Job Controls            AMGCUPIX   AMDCUPIX

Type overrides, press Enter.

 ENU  English uppercase and lowercase
 Job description  . . . .  Systems and Versions . . . . . .

 CHT  TRADITIONAL CHINESE DBCS (ROC)
 Job description  . . . .   系統和版本  . . . . . . . . . .

                                                                     BOTTOM
F3=Exit  F12=Cancel
```

Figure 8-9: Job Controls language override screen

5 To override the job description text, type the new text and press Enter.

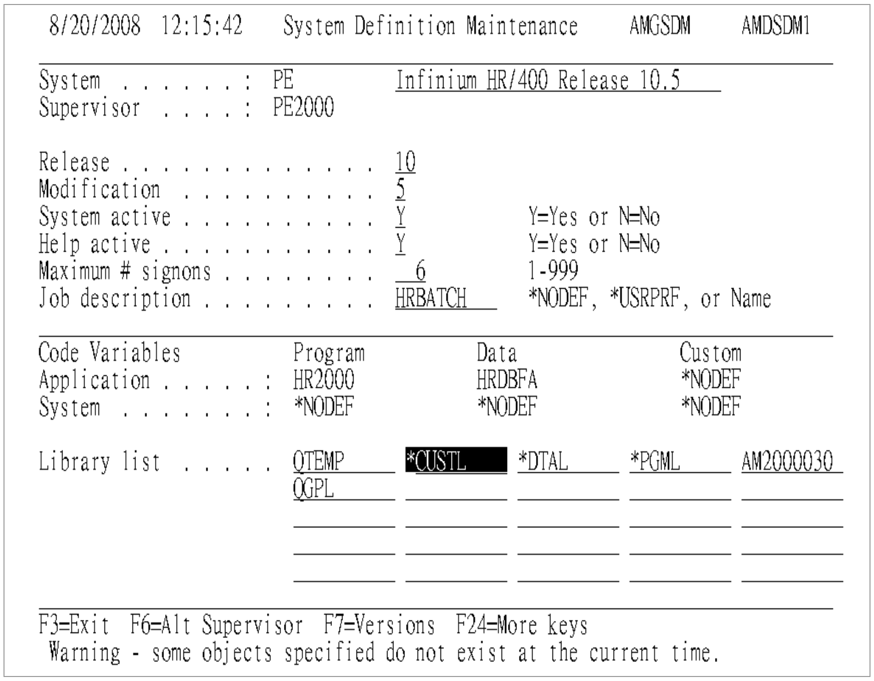
Working with menu language overrides

Complete the steps below to manually add menu language overrides.

1 From the Infinium AM main menu, select *Systems and Versions*.

2 Select a version with **5**.

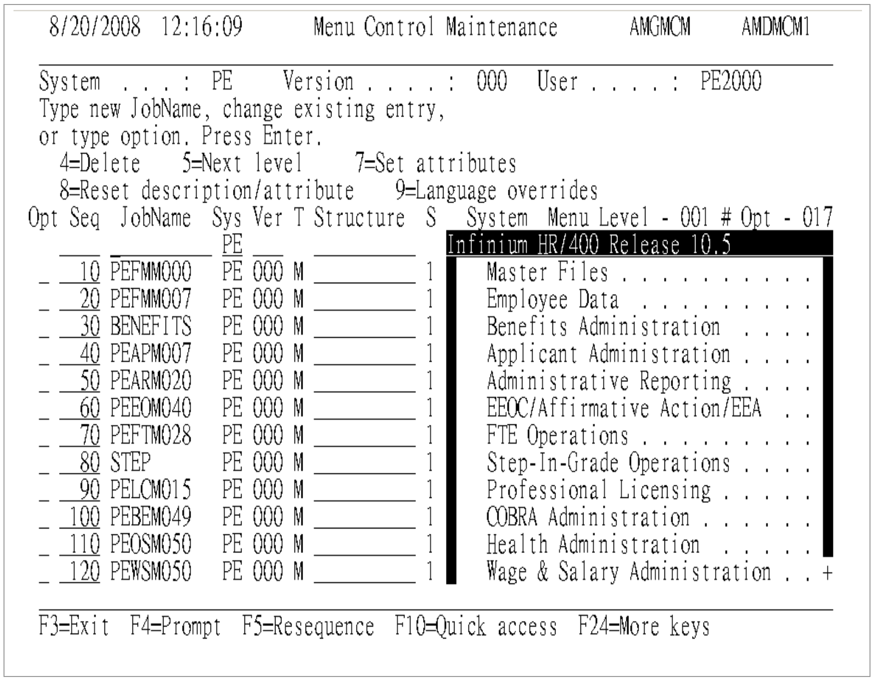
3 Press Enter. The system displays a screen similar to Figure 8-10.



```
 8/20/2008  12:15:42    System Definition Maintenance     AMGSDM     AMDSDM1

System  . . . . . . :  PE          Infinium HR/400 Release 10.5
Supervisor  . . . . :  PE2000

Release . . . . . . . . . . . . . .  10
Modification  . . . . . . . . . . .  5
System active . . . . . . . . . . .  Y          Y=Yes or N=No
Help active . . . . . . . . . . . .  Y          Y=Yes or N=No
Maximum # signons . . . . . . . . .    6        1-999
Job description . . . . . . . . . .  HRBATCH    *NODEF, *USRPRF, or Name

Code Variables          Program          Data             Custom
Application . . . . . :  HR2000          HRDBFA           *NODEF
System  . . . . . . . :  *NODEF          *NODEF           *NODEF

Library list  . . . . .  QTEMP     *CUSTL     *DTAL     *PGML     AM2000030
                         QGPL

F3=Exit  F6=Alt Supervisor  F7=Versions  F24=More keys
 Warning - some objects specified do not exist at the current time.
```

Figure 8-10: System Definition Maintenance screen

4 Press F11. The system displays a screen similar to Figure 8-11.

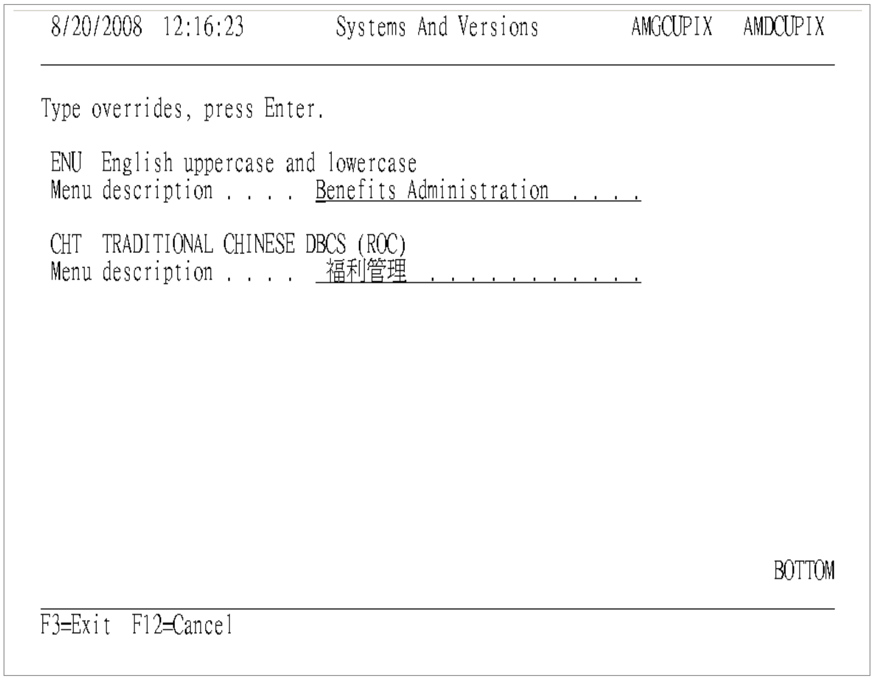


```
 8/20/2008  12:16:09       Menu Control Maintenance       AMGMCM     AMDMCM1

System  . . . :  PE     Version . . . . :  000   User . . . . :  PE2000
Type new JobName, change existing entry,
or type option. Press Enter.
  4=Delete    5=Next level     7=Set attributes
  8=Reset description/attribute    9=Language overrides
Opt Seq  JobName  Sys Ver T Structure  S   System  Menu Level - 001 # Opt - 017
    ____ _________ PE ___   _________     Infinium HR/400 Release 10.5
 _   10 PEFMM000   PE 000 M _________  1      Master Files . . . . . . . . . .
 _   20 PEFMM007   PE 000 M _________  1      Employee Data  . . . . . . . . .
 _   30 BENEFITS   PE 000 M _________  1      Benefits Administration  . . . .
 _   40 PEAPM007   PE 000 M _________  1      Applicant Administration . . . .
 _   50 PEARM020   PE 000 M _________  1      Administrative Reporting . . . .
 _   60 PEEOM040   PE 000 M _________  1      EEOC/Affirmative Action/EEA  . .
 _   70 PEFTM028   PE 000 M _________  1      FTE Operations . . . . . . . . .
 _   80 STEP       PE 000 M _________  1      Step-In-Grade Operations . . . .
 _   90 PELCM015   PE 000 M _________  1      Professional Licensing . . . . .
 _  100 PEBEM049   PE 000 M _________  1      COBRA Administration . . . . . .
 _  110 PEOSM050   PE 000 M _________  1      Health Administration  . . . . .
 _  120 PEWSM050   PE 000 M _________  1      Wage & Salary Administration . . +

 F3=Exit  F4=Prompt  F5=Resequence  F10=Quick access  F24=More keys
```

Figure 8-11: Menu Control Maintenance screen

5 Type **9** in the *Opt* field to change the language for a menu description.

6 Press Enter. The system displays a screen similar to Figure 8-12.



```
 8/20/2008  12:16:23        Systems And Versions        AMGCUPIX   AMDCUPIX


Type overrides, press Enter.

 ENU  English uppercase and lowercase
 Menu description . . . .  Benefits Administration  . . . .

 CHT  TRADITIONAL CHINESE DBCS (ROC)
 Menu description . . . .  福利管理  . . . . . . . . . . .










                                                                      BOTTOM

F3=Exit  F12=Cancel
```

Figure 8-12: Menu Control Maintenance message ID window

7. To override the menu description text, type the new text.
8. When you have finished, press F3 to exit and save.
9. Repeat this task until you finish overriding menu options.
10. Exit and save your changes.

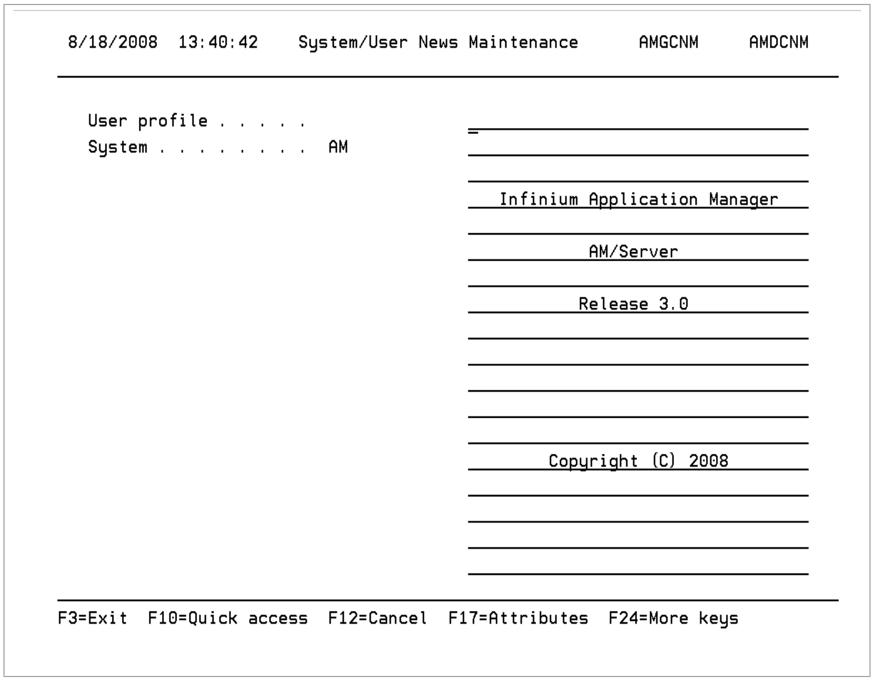
Working with news language overrides

Complete the steps below to manually add news language overrides.

1 From the Infinium AM main menu, select *News*.

2 Specify a user ID and system.

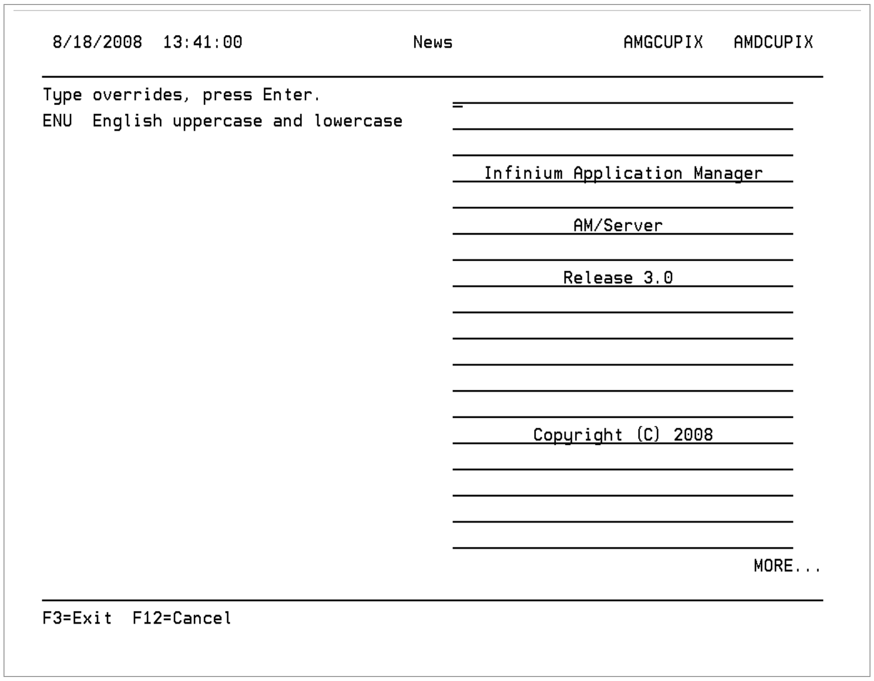
3 Press Enter. The system displays a screen similar to Figure 8-13.



```
 8/18/2008  13:40:42    System/User News Maintenance      AMGCNM     AMDCNM

   User profile . . . . .
   System . . . . . . . .  AM

                                          Infinium Application Manager

                                                   AM/Server

                                                  Release 3.0

                                              Copyright (C) 2008

F3=Exit  F10=Quick access  F12=Cancel  F17=Attributes  F24=More keys
```

Figure 8-13: System/User News Maintenance screen

4 Press F9. The system displays a screen similar to Figure 8-14.

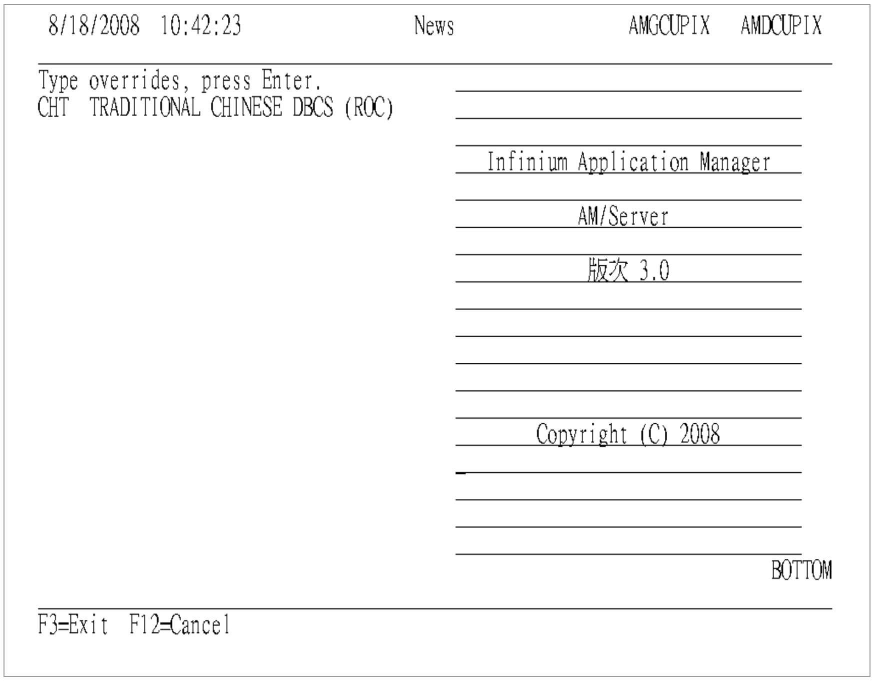


```
 8/18/2008  13:41:00                 News                 AMGCUPIX   AMDCUPIX

Type overrides, press Enter.
ENU  English uppercase and lowercase

                                             Infinium Application Manager

                                                      AM/Server

                                                     Release 3.0






                                                 Copyright (C) 2008




                                                                        MORE...

F3=Exit  F12=Cancel
```

Figure 8-14: System/User News Maintenance screen

5 Review and update the text as necessary.

6 Press Page Down to update the news for the alternate language. The system displays a screen similar to Figure 8-15.



```
 8/18/2008  10:42:23                 News                 AMGCUPIX   AMDCUPIX

Type overrides, press Enter.
CHT  TRADITIONAL CHINESE DBCS (ROC)

                                             Infinium Application Manager

                                                      AM/Server

                                                      版次 3.0






                                                 Copyright (C) 2008




                                                                        BOTTOM

F3=Exit  F12=Cancel
```

Figure 8-15: System/User News Maintenance screen

7. To override the news text, type the new text.

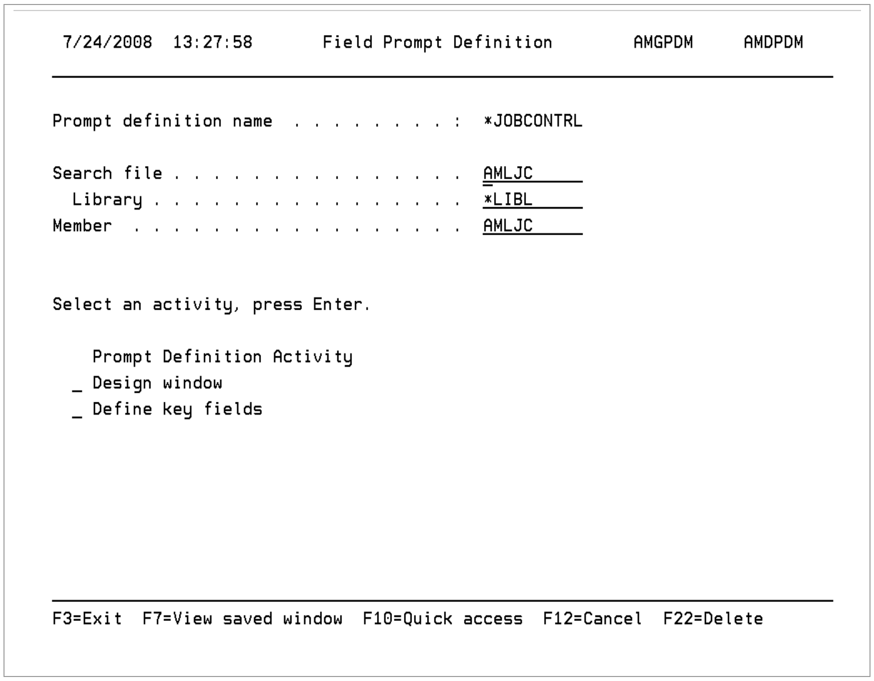
8. Exit and save your changes.

Working with field prompt language overrides

Complete the steps below to manually add field prompt language overrides.

1 From the Infinium AM main menu, select *Field Prompts*.

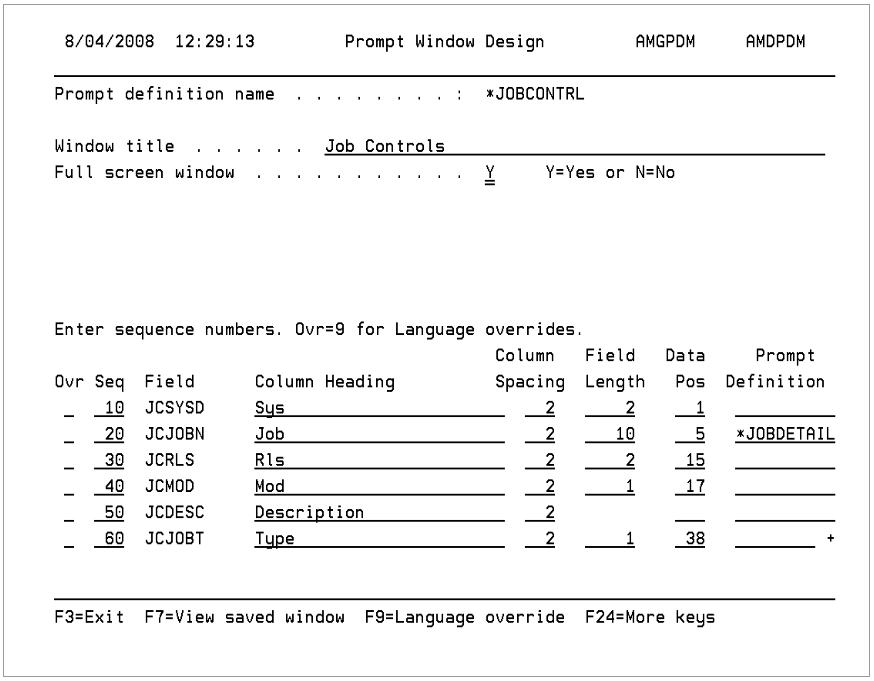
2 Specify a field prompt. The system displays a screen similar to Figure 8-16.



```
 7/24/2008  13:27:58       Field Prompt Definition        AMGPDM     AMDPDM

Prompt definition name  . . . . . . . . . :  *JOBCONTRL

Search file . . . . . . . . . . . . . . .  AMLJC
  Library . . . . . . . . . . . . . . . .  *LIBL
Member  . . . . . . . . . . . . . . . . .  AMLJC


Select an activity, press Enter.

    Prompt Definition Activity
  _ Design window
  _ Define key fields






F3=Exit  F7=View saved window  F10=Quick access  F12=Cancel  F22=Delete
```

Figure 8-16: Field Prompt Definition screen

3 Select *Design Window* and press Enter. The system displays a screen similar to Figure 8-17.



```
 8/04/2008  12:29:13         Prompt Window Design         AMGPDM     AMDPDM

Prompt definition name  . . . . . . . . . :  *JOBCONTRL

Window title  . . . . . .  Job Controls
Full screen window  . . . . . . . . . . .  Y     Y=Yes or N=No




Enter sequence numbers. Ovr=9 for Language overrides.
                                            Column   Field   Data    Prompt
Ovr Seq  Field      Column Heading          Spacing  Length   Pos  Definition
 _   10  JCSYSD     Sys                        2        2      1
 _   20  JCJOBN     Job                        2       10      5   *JOBDETAIL
 _   30  JCRLS      Rls                        2        2     15
 _   40  JCMOD      Mod                        2        1     17
 _   50  JCDESC     Description                2
 _   60  JCJOBT     Type                       2        1     38             +


F3=Exit  F7=View saved window  F9=Language override  F24=More keys
```

Figure 8-17: Prompt Window Design screen

Use this screen to override column headings and window text.

To override the column heading text, complete the steps below.

a Type **9** in the Opt field to change the language for a column heading.

b Press Enter. The system displays the language override screen.

c To override the current column heading text, type the new text and press Enter.

To override the window title text, complete the steps below.

a Press F9. The system displays the language override screen.

b To override current window title text, type the new text and press Enter.

4 Exit and save your changes.