Table of Contents

Backing Up Files

The following files will need to be backed up periodically as deemed necessary by your Open/A site administrator. This will ensure that you have a back up copy in case any one of these files become lost or corrupted.

OPENA/CONFIG/=
OPENA/CONFIG
<all client configuration files>

Getting Started

The first step in using Open/A is to install the mainframe software. Follow the steps as documented in chapter one under “Initial Installation”.

The second step is to install the Open/A PC software. In chapter two, read the sections “Initial Installation”, “Custom Installation” and “Installation Switches” before installing the Open/A software. Follow the steps as documented in the section “Initial Installation” and apply steps from the sections “Custom Installation” and “Installation Switches” if desirable.

The third step is to install the Administration program. Follow the steps as documented in chapter two, section “Installation of the Administration Program”.

Now, you are ready to run the administration program and create client configuration files. Once the client configuration file(s) is created, you will be able to access data via PC application software. Refer to chapter three for information relative to using the administration program.

Chapter 1

Installation of Open/A Mainframe Component For the A-Series

Introduction

The steps to install the mainframe component of the Open/A software are documented below. The installation process is simplified for ease of use. Basically, the necessary software is automatically copied to the mainframe; then, either the COMS direct Window program is set up for communication via Infoconnect or CTCBRIDGE, or you may use Open/A to communicate via HLCN, TCP/IP, Telnet, Upstanding NTP, HFS or the APIS transport.

Initial Installation

To install the Open/A software on your mainframe, you will need to follow the steps below:

1. Copy the files on the distribution tape to your mainframe. Copy the files to the usercode and packname of your choice.

Note: The files on the distribution tape are under the usercode (OPENA).

2. WFL MODIFY the program "(OPENA)OBJECT/OPENA/ADMIN ON PACK" if the Open/A administrator does not wish to allow users to create their own client configuration files. The syntax is as follows:

```
WFL MODIFY (OPENA)OBJECT/OPENA/ADMIN ON PACK; SW1=TRUE
```

Note: Users will be required to log onto the host under a privileged A-Series usercode in order to use the administration program if the file "OBJECT/OPENA/ADMIN" has SW1=TRUE.

3. Set up the COMS direct window program **if you are not using direct HLCN or if you are not using direct TCP/IP**. Refer to Chapter 4 at this time if you are using direct HLCN; refer to Chapter 5 if you are using direct TCP/IP; otherwise, refer to chapter 6 for the COMS direct window program information.

NOTE: It is not necessary to set up the COMS direct window program or set up stations if you are using direct HLCN or direct TCP/IP.

4. The mainframe installation is now complete. If you have already installed the PC software, you are ready to use Open/A.

Chapter 2

Installation of Open/A Windows Component

Introduction

The steps to install the Windows component of the Open/A software are documented below. The files on the diskettes are compressed and must be installed on your hard drive. In addition, the Open Database Connectivity software must be installed and configured with the supplied software.

The setup program will determine if you are running Windows(16-bit) or Windows95 (32-bit) and install the proper driver accordingly. You may use 16-bit applications with the 32-bit driver.

Initial Installation

In order to install the Windows driver component, you will need to be running Windows. Once Windows is running, follow the steps below for driver installation. **For setup switch information, refer to the Installation Switches section.**

1. Place the first distribution floppy into the appropriate drive on your PC.

2. From Program Manager, select the “File” menu option; or, from Windows 95, click on the Start button on the taskbar. From the drop down list, select the “Run” menu option.

3. In the “Command Line” box, key in the following statement for a normal installation, which is the default installation type.
   ```
   <drive letter>:\setup.exe
   ```
   This command will run the program “setup.exe” on the distribution diskette. You will see an “Initializing Setup” screen.

4. After setup initializes, you will see the “Welcome” screen.

5. Click on “Next” to continue; you will be presented with the setup type screen. You may choose “Cancel” to cancel at this time if you wish. You may choose “Back” to go back to the previous screen.

6.. On the “Setup Type” screen, Open/A setup provides three choices for the setup type; for a local install, choose the option “Typical”; for a network install, choose “Network”; to link to program files after a network install has been completed, choose “Link”. Click on “Next” to continue. You may choose “Back” to go back to the previous screen; choose “Cancel” to exit setup.

NOTE: Once a “Link” install has been done, a new network install that specifies the same directory will cause all users to automatically link to the new version.

7. You will be presented with the “Specify Location” screen. By default, the Open/A driver will be copied to the specified directory “Windows\System” . Click “Next” to continue. Click “Browse” to select a different directory. Click “Back” to go back to the previous screen. Click “Cancel” to exit setup.

8. You will be presented with the “Select Installation Components” screen. Select the components to install accordingly. Clear any component you do not wish to install. **You may wish to select applicable transport(s) to install at this time; otherwise, all transports are installed; click on the “check mark” beside “Transports” and you will see a list of available transports from which to choose.**

Click on “Next” to continue. Click on “Back” to go back to the previous screen; click on “Cancel” to exit setup.

NOTE: You may add a transport anytime by running the “setup.exe” program and choosing the appropriate transport.

If previous data sources **do not exist**, you will be presented with a screen in order to specify a data source name and description. You may want to specify a data source name that can be easily recognized by end users. This data source name is used to refer to a given connection to the A-Series; it will be used to access the A-Series data from ODBC applications such as Microsoft Access and PowerBuilder. The description may be any text which will uniquely describe this datasource.

NOTE: You may have more than one data source set up.

Specify a file name for the client configuration file, including the A-Series usercode and packname. The Client Configuration File is a file on the A-Series which specifies all of the configuration options under which this connection will operate. Included in this configuration file are the names of the datasets to which this particular connection may have access. This file is created and maintained by a Windows based administration program; you will find instructions in the Administration section regarding how to create and maintain this file via the administration program.

NOTE: The client configuration file should be public; however, if security is a concern, you may make it private with a guard file specified on it, or put the configuration file under the usercode of the user.

The Userid and Password may be any valid A-Series usercode and password. You will need to specify a usercode that has access to any database(s) that you wish to use. This will be the usercode under which you will be actively running when accessing data via Open/A.

The Communications Provider and Connection Name will need to refer to he path which will attach the datasource to the host communications software. **Refer to chapter 4, section "Windows PC Setup" if you are using direct HLCN; if you are using direct TCP/IP, refer to chapter 5, section "Windows PC Setup"; otherwise, refer to chapters 6 and 7, "Setting up COMS Direct Window Program and Stations" and "Transport Specifics".**

9. On the "Setup" screen, you will see a message indicating that the files on the distribution diskette are being copied to your hard drive. You will receive a message indicating that setup needs the next diskette. At this time, insert the second driver diskette and click on "OK". You may click "Cancel" to exit setup.

10. You will be presented with an "Information" screen which will inform you that setup is complete. Click "OK".

NOTE: The setup program will inform you if it is necessary to restart Windows; otherwise there is no need to restart Windows.

The installation process added an ODBC icon to your Control Panel window, if necessary. If you ever need to configure, add, or delete data sources or drivers, you will need to use this icon. You will be presented with a configuration screen which will allow you to specify a datasource name, a description, the client configuration file name(including usercode and packname), usercode/password, transport, and connection name.

The Windows software is now installed and may be used if the mainframe software has been installed.

Installation Switches

Open/A provides the usage of the following switch names for the driver installation. The actual switch name is specified at run time on the <drive letter>:\setup.exe statement. For example, valid syntax is as follows:

```
<drive letter>:\setup.exe  <switch name>
```

NOTE: The switch name references the actual term which describes the type of setup you are performing. (ie: Typical, Network, or Link)

Three switch modes exist:

1. **Typical** - This is the default mode. This mode of installation copies files from the floppy diskette to the local workstation.

2. **Network** - This type of installation copies all shared software that normally goes into the Windows\System directory to a network shared Windows directory. This type of setup will need to be run by the network administrator each time an Open/A setup disk is received.

3. **Link** - Users will need to run setup and specify the same directory that was specified in step 2. As a result, links will be made to files installed during the network setup.

NOTE: Once step 3 has been done, a new network install that specifies the same directory will cause all users to automatically link to the new version.

Installation of the Administration Program

The Open/A Windows based administration program is shipped on separate diskettes. This is to prevent unauthorized users from running the administration program. If you plan to run the administration program, you will need to install it at this time. Since it uses the same communications libraries, it must be installed on a machine which already has the Open/A driver installed on it. The administration program provides the usage of three switches; these switches are specified by name and have the same functionality as the switches used in the driver setup.

To install the administration program, follow the steps below:

1. From Program Manager, select the “File” menu option; or, from Windows 95, click on the Start button on the taskbar. From the drop down list, select the “Run” menu option.

2. In the “Command Line” box, key in the following statement:

   ```
   <drive letter>:\setup.exe
   ```

   This command will run the installation program “setup.exe” on the distribution diskette.

3. You will be presented with an “Initializing Setup” screen, followed by a “Welcome” screen. Click on “Next” to continue or click on “Cancel to exit setup.

4. You will be presented with the “Setup Type” screen. Open/A setup provides three choices for the setup type; for a local install, choose the option “Typical”; for a network install, choose “Network”; to link to program files after a network install has been completed, choose “Link”. Click on “Next” to continue. You may choose “Back” to go back to the previous screen; choose “Cancel” to exit setup.

5. You will be presented with the “Specify Program Files Location” screen. Open/A administrator will be installed in the specified directory; if this directory is correct, click on “Next” to continue. Choose “Browse” to select a different directory. Click on “Back” to go back to the previous screen. Click on “Cancel” to exit setup.

5. The next screen will allow you to specify the location of the “Windows\System” directory.

7. You will be presented with the “Select Installation Components” screen. Select the components to install accordingly. Clear any

component you do not wish to install. **You may wish to select applicable transport(s) to install at this time; otherwise, all transports are installed.**

Click on “Next” to continue. Click on “Back” to go back to the previous screen; click on “Cancel” to exit setup.

8. The “Setup” screen will display a message indicating that the files are being copied. You will be presented with another screen which indicates that setup needs the next Open/A administration diskette. Insert the next administration diskette and click “OK” to continue; click on “Cancel” to exit setup.

9. The “Information” screen will inform you that setup is complete. You may now use the administration program.

   NOTE: The setup program will inform you if it is necessary to restart Windows; otherwise there is no need to restart Windows.

A program icon will be created in “Start Programs” by the name “Open A”. Once you have installed the administration program, you are ready to start configuring the A-Series Client Configuration files; the configuration file is actually specified as indicated in step eight above, section titled “Initial Installation”. For a discussion of client file configuration, refer to chapter 3, “Open/A Administration Program”.

Administration Setup Switches

Open/A provides the usage of the following three switch names for the administration program. The default mode is no switch is necessary. These switch names are specified as indicated below:

```
<drive>:\setup.exe  <switch name>
```

Three switch modes exist:

1. **Typical -** This is the default mode. This mode of installation copies files from the floppy diskette to the local workstation.

2. **Network -** This type of installation copies all shared software that normally goes into the Windows\System directory to a network

shared Windows directory. In addition, all other setup files will be copied into a user specified setup directory on the network. This type of setup will need to be run by the network administrator each time an Open/A setup disk is received.

3. **Link -** Users will need to run setup from the user specified setup directory indicated in step 2.

NOTE: Once step 3 has been done, a new network install that specifies the same directory will cause all users to automatically link to the new version.

Chapter 3

Open/A Administration Program

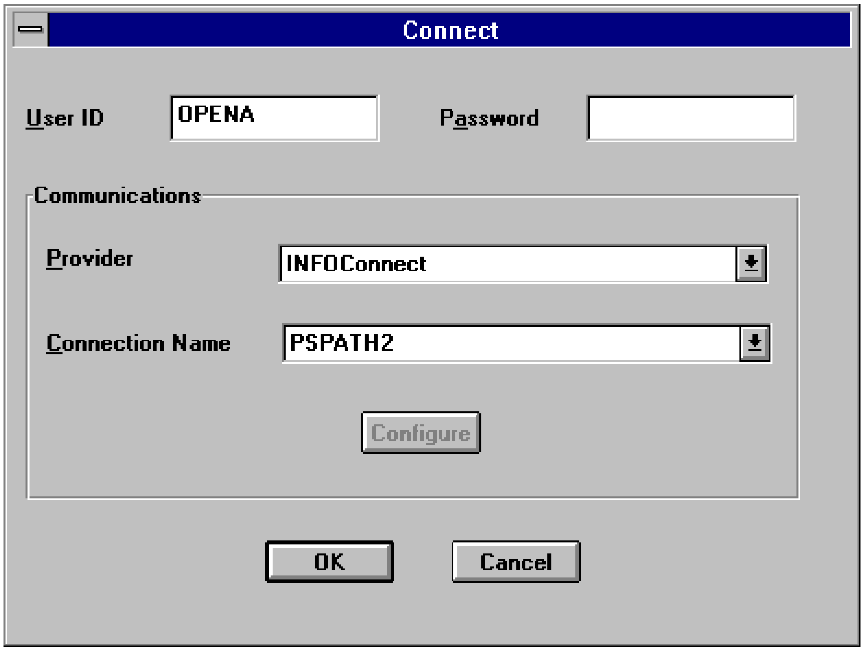
Introduction

The Open/A Administration program runs under Microsoft Windows and is used to control all aspects of the Open/A operation. The main functions of the administration program are the ability to enter a software license key, maintenance of tables in Open/A client configuration files, maintenance of stored procedures, application security, ability to set limits, ability to set global default options, and predefined join definitions. As discussed in chapter 2, "Installation of Open/A Windows Component", the client's ODBC data source specifies the name of a client configuration file on the host. This configuration file contains a list of all DMSII structures that the user may access. Each user may be given his own client configuration file or groups of users may share a client configuration file. The creation and maintenance of these configuration files via the administration program are discussed in the following sections.

Installation and Configuration

The installation of the Administration program is discussed in chapter 2, "Installation of Open/A Windows Component", section " Installation of Administration Program". To paraphrase this process, the Open/A distribution diskettes contain the Administration installation program, "setup.exe", which is run from Program Manager under Windows, or you may choose "run" from the Win95 Start menu. The administration program also provides the usage of switch names which are discussed in chapter 2. **This program is not meant to be used by every user; consequently, you may want to remove it from any diskettes that you distribute to your users.**

NOTE: In order to function correctly, the Administration program must be run from a workstation that already has the Open/A ODBC driver installed.



User ID: OPENA

Password:

Provider: INFOConnect

Connection Name: PSPATH2

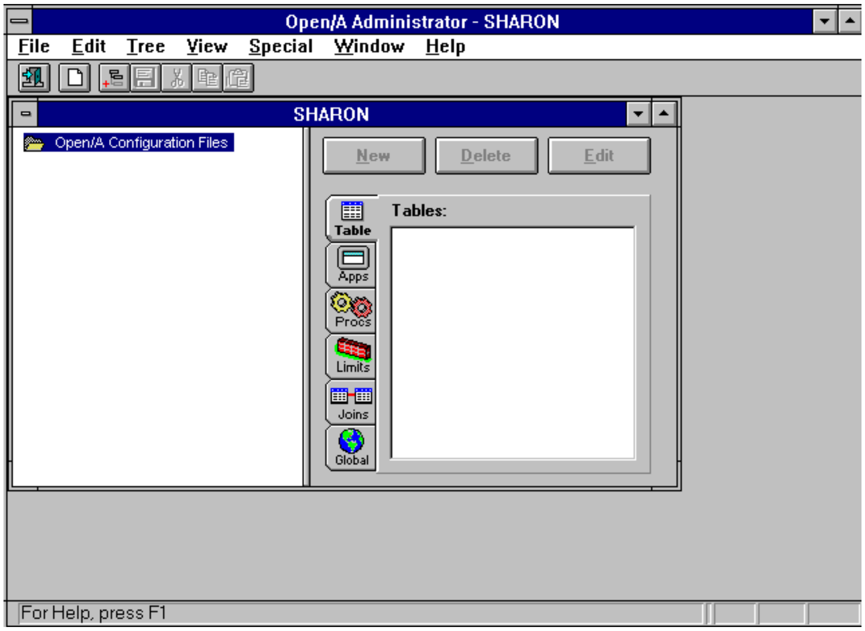
Open/A Administration Connection Screen

To run the administration program, double click on the Open/A administration icon; or, from the Win95 start menu, choose “Programs” and then choose the “Administrator” icon accordingly. You will be presented with a dialog box which is very similar in appearance to the Open/A ODBC data source setup dialog box. Refer to the figure above. You will need to enter your A-Series usercode/password in the proper location within the dialog box. **The usercode and password which is specified here will be used to log the administration user onto the host.** If you are a privileged usercode, you will be able to create and modify client configuration files for all users on your system from a single session.

NOTE: Users will be required to log onto the host under a privileged A-Series usercode in order to use the administration program if the file “OBJECT/OPENA/ADMIN” has SW1 equal to TRUE.

A section also exists for you to specify the communications channel to your host. The communications provider and connection name will need to reference a station which has been set up to use the Open/A ODBC driver (refer to chapter 2, “Installation of Open/A Windows Component” for a discussion of these settings).

At this time, the Administration program does not share a communications channel with the ODBC driver; consequently, all open Open/A ODBC applications on your station will need to close their connections. After supplying these pieces of information, you will need to click on “OK” and you will be logged onto the host.



Open/A Administration Screen

The Open/A administration window will be split into two parts; the left side of the window will reference client configuration files and the right side of the window references tables, applications, stored procedures, limits, joins, and global default settings. Henceforth, the left side of the Open/A window will be referred to as the Open/A client configuration file window and the right side will be referred to as the Open/A table window.

There are two ways in which you may select operations to perform while in the Open/A administration program. The first method is by using the standard Windows Menu bar. The second method is the usage of the right mouse button. While the cursor is over the Administration window, you may click the right mouse button and you will be presented with a list of your currently available options. This method will be a little quicker than using the Windows Menu bar.

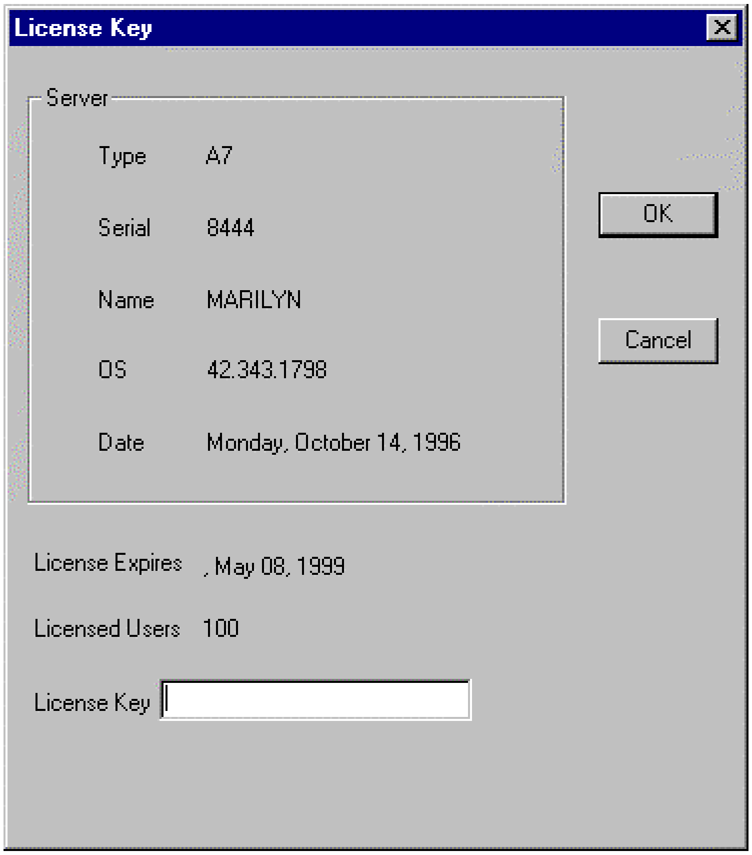
The various sets of operations which may be performed by the Administration program are discussed below.

License Key

The license key option allows you to extend your trial or update your license date periodically. You will receive your license key from Software Clearing House accordingly. The license key option is accessible as follows:

1. Click on “Special” on the menu bar.

2. Choose the “License Key” option from the drop down list. You will be presented with a “License Key” dialog box window which will display information regarding your A-Series; such as machine type, serial number, host name, MCP level, and current date. At the bottom of the window, you will see your license information.



Server

Type	A7
Serial	8444
Name	MARILYN
OS	42.343.1798
Date	Monday, October 14, 1996

License Expires	, May 08, 1999
Licensed Users	100
License Key	

License Key Dialog Box

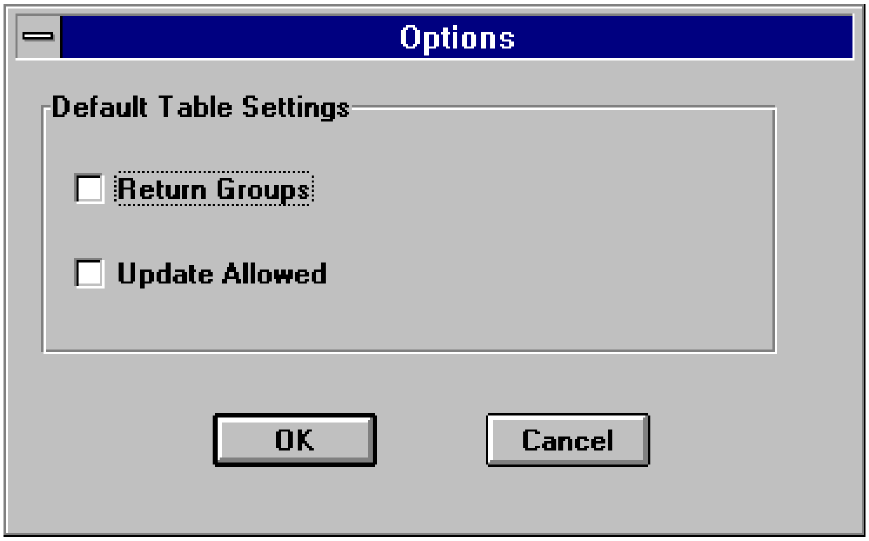
3. Enter your new license key in the box labeled “License Key”, located at the bottom of the dialog box window. Click “OK”. You may select the “Cancel” option if you do not wish to enter a license key at this time.

Default Options

Under the “Special” option on the menu bar, there is a drop down option called “Default Options”. This option will allow you to globally specify whether all programs in the available table list are “updateable” programs and whether or not group fields will be returned to the calling program. Note that these options are not applicable to existing tables in the table list; they apply to new tables that are added to the table list. You will need to edit any existing table(s), if applicable, and set the option(s) accordingly. The default value for these options is off (not set).

1. Click on “Special” on the menu bar.

2. Choose the option “Default Options” from the drop down list.



Default Table Settings

□ Return Groups

□ Update Allowed

Default Options Dialog Box

3. Place an “x” in the appropriate check box(es) for the options(s) you wish to set. You may select the “Cancel” option if you do not wish to set options at this time.

Preferences

Session preferences allow you to specify that you wish to confirm client configuration file deletions, you may specify that you wish to automatically save the client configuration files, or you may specify that new tables will be created from a COBOL source.

If the “Confirm Configuration File Delete” option is set, you will be prompted with a confirmation message asking you if you wish to delete the file; otherwise, you will not receive the confirmation message. This option is set by default.

If the “Automatically Save Configuration Files” option is set, you will be prompted with a message which will ask you if you wish to save the changes to your client configuration file.

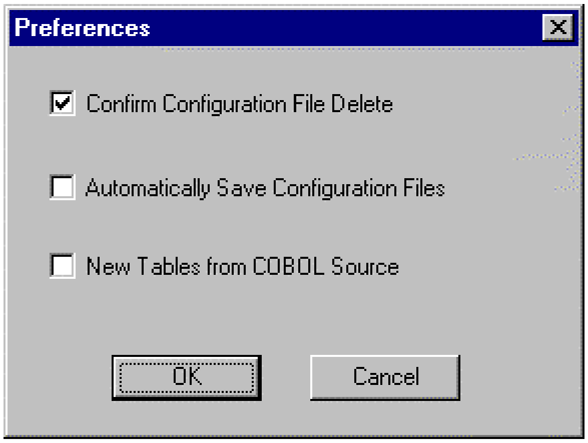
If the “New Tables from COBOL Source” is set, you will be presented with a dialog box which will allow you to specify a Cobol source program name, data file name, table name, and FD or 01processing information, and optionally, an access library name. This option is to be used for flat files and KIOII files; when set, Open/A automatically generates table descriptions for these files; thereby, eliminating the need to manually define the layouts of these file types.

Session preferences are accessible as follows:

1. Click on “Special” on the menu bar.

2. Choose “Preferences” from the drop down list.

3. Place an “X” in the appropriate box(es) for the option(s) you wish to set. Click on “OK”. You may select the “Cancel” option if you do not wish to set the preference options at this time.



☑ Confirm Configuration File Delete

□ Automatically Save Configuration Files

□ New Tables from COBOL Source

Preferences Dialog Box

Environment Command

The environment command allows you to send commands to the environment manager via a dialog box; thereby allowing you to control the environment manager. Refer to chapter 11 regarding additional information on environment logging.

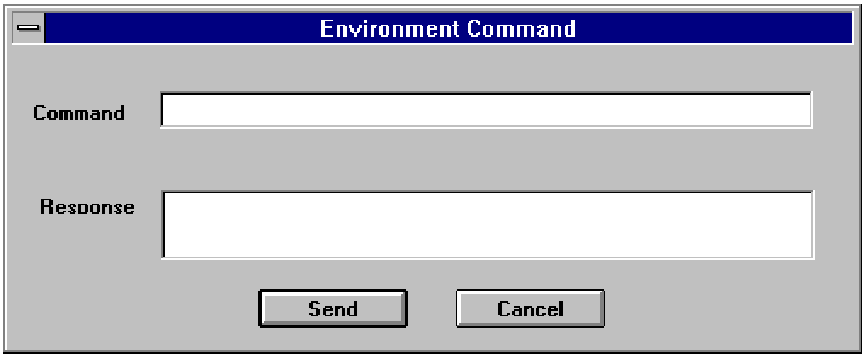
To access the environment command:

1. Click on “Special” on the menu bar.

2. Choose “Environment Command” from the drop down list.

3. Key in a valid command and click on “Send” or press the “Enter” key on the keyboard. Refer to chapter 11, “Logging Feature” for information regarding the valid environment commands.

You may “Cancel” if you do not wish to send any commands to the environment manager at this time.



Environment Command Dialog Box

Client Configuration File Management

The client configuration file list is located on the left side of the Open/A window. The client configuration file management will be maintained via a tree structure similar to File Manager. As you add client configuration files to the list, you will notice the folder and file type structure that will be created. Note that the client configuration file location on the tree is based upon the node that is selected before you actually "add" the file(s).

Initially, there will be no client configuration files in the list; thus, you will need to "Add" a client configuration file. To add a client configuration file, click on "File" on the menu bar; then choose "Add Configuration File". You will be presented with a dialog box which will enable you to specify the name of the client configuration file; **you will need to include the usercode and packname.** Once you have named your client configuration file, you will need to proceed to the Table Selection dialog box which will allow you to specify your description file name, choose a database, and select structure names for entry into the available table list. Refer to the "Table Selection" section below for information regarding the table selection process.

On the drop down list of the "File" menu bar option, you may use the "Save Configuration File" option to save your client configuration file; or you may use the icon on the toolbar.

If you change to a different client configuration file or exit from the administration program with any unsaved changes, you will be prompted with a message which will ask you if you wish to save the changes to your client configuration file; you will need to respond accordingly. There is an option available which will allow you to indicate that you wish to automatically save the client configuration file; to locate this option, click on "Special" on the menu bar, then click on "Preferences".

If a client configuration file already exist on your host, you may use the "Add Configuration File" option to add it to your tree list. Then, you will be able to add new tables, edit tables, delete existing tables, etc. Again, note that the location of the client configuration file on the tree is based upon the node that is selected before you actually "add" the file.

Likewise, to edit an existing client configuration file, click on the desired client configuration file name in the tree list; you will be presented with the corresponding table information in the table window. You will now be able to add new tables, edit tables, delete existing tables, etc.

On the drop down list of the "File" menu bar option, you may use the "Delete Configuration File" option to physically delete a client configuration file.   You will be prompted with a confirmation message asking you if you wish to to delete the file; you will need to respond accordingly. There is an option available which will allow you to indicate that you do not wish to receive this confirmation message; to locate this option, click on “Special” on the menu bar, then click on ”Preferences”.

In order to remove the client configuration file reference from the tree structure only, you will need to hold down the “ctrl” key on the keyboard and depress the “Delete” key. Note that the physical file will still exist on the A-Series.

To rename the title of the client configuration file, you may use the “Rename Configuration File” option on the drop down list of the “File” menu bar.

If you wish to move a client configuration file to a different disk pack and have the administration program reference the client configuration file on the new pack, you will need to physically copy the client configuration file to the new disk pack, delete the original client configuration file, and “add” the client configuration file on the new pack via the “Add Configuration File” option in the drop down list of the “File” menu option.

You may use cut, copy, and paste with the client configuration files. After a client configuration file has been chosen, you may use “Cut” to remove it; cut actually performs a “copy” of the client configuration file to the clipboard and then removes it. You may “paste” this client configuration file back into your configuration file list by choosing the “paste” option and providing a configuration file name. “Copy” copies the configuration file to the clipboard. You may then “paste” this client configuration file back into your configuration file list by choosing the “paste” option and providing a configuration file name.

Table Selection

The primary usage of the Open/A Administration program is the setup and maintenance of the specific DMSII structures that users are allowed to access. These DMSII structures are specified as indicated below.

1. Click on “Table” on the Open/A table window.

2. Choose the “New” button. You will be presented with a “Table Selection” dialog box.

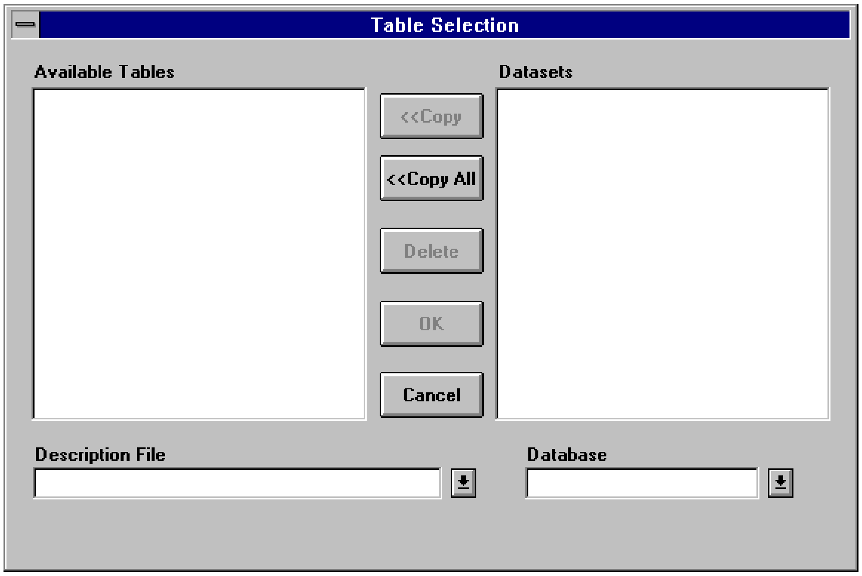
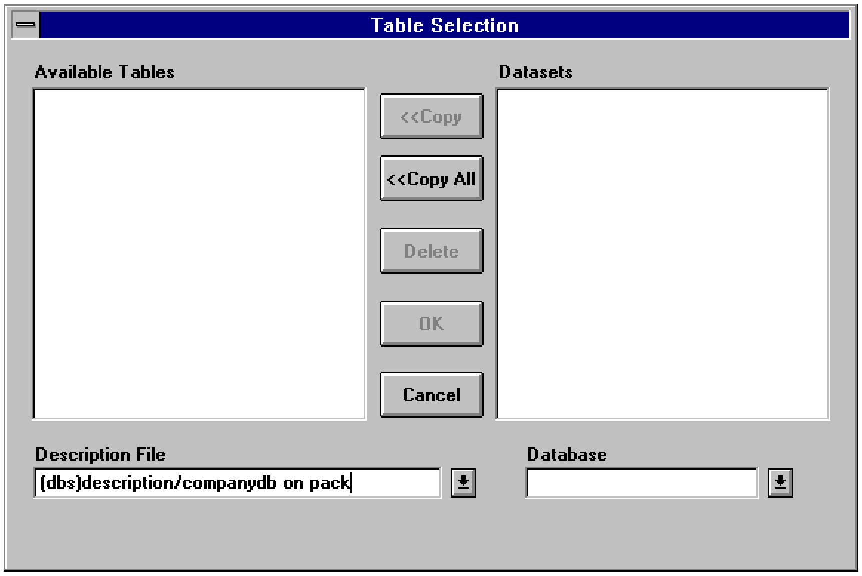


Table Selection Dialog Box

3. To add one or more tables to the list, you will first need to enter the name of a DMSII description file that you wish to access. **The description file name will need to contain a usercode and a pack name**. Once a description file has been accessed, it will remain in the "Description File" drop down list each time you run the Administration program. Then, you may choose the description file name from the drop down list or you may enter it in the proper location in the dialog box. If a description file is not found on the A-Series host machine, a not found message will appear; in addition, the corresponding entry in the "Description File" drop down list will be removed, if applicable.

Anyone running the Administration program will see all the description file names which have been specified at any given time; including those specified by any other administrator.

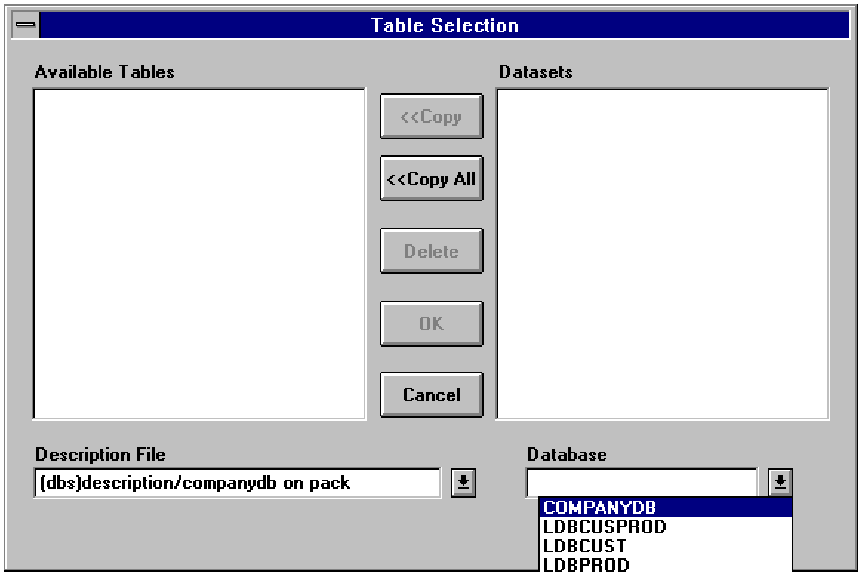


Description File: (dbs)description/companydb on pack

Description File Selection

4. After specifying the description file, you will need to click on the “Database” drop down list. Choose a database from the list.

NOTE: While the Administration program is retrieving a list of databases, physical and logical, in the given description file, you may see the hourglass cursor momentarily.



Description File: (dbs)description/companydb on pack

Database:
COMPANYDB
LDBCUSPROD
LDBCUST
LDBPROD

Database Selection

5. Select datasets, set(s), subset(s), and remaps to be added to the list of available tables by clicking on the name of the desired structure and then clicking on the “Copy” button. You may also double-click on a single structure name to move it to the available table list. For multiple structure selection, hold down the shift or ctrl key as you make your selections; to select all the tables, you may use the “copy all” button. If the same dataset is selected more than once, a dialog box will appear which will indicate that a duplicate table name exist.

NOTE: After selecting a database, you will see the hourglass again as the list of datasets, sets, and subsets is built and displayed.

Steps 3 thru 5 may be repeated for multiple DMSII databases per client configuration file.

Dataset names will appear in black, bold print; however, if the dataset has been declared as updateable, it will appear in red print. If a dataset name is not completely visible in the dataset list or the available table list, you may use the arrow keys to scroll to the right to view the rest of the data.

If a dataset or remap table is used in a query, Open/A will use any of the available sets for that table for query criteria evaluation. If a set or subset table is used in a query, Open/A will always use that particular set/subset to read the dataset, no matter what the criteria may be. The ability to use a set/subset in a query guarantees that a user will always use a particular set/subset to read the dataset.

Embedded datasets, for **inquiry** purposes, may be selected and added to the available table list. All possible combinations of embedded structures will be listed in the dataset list with the following naming convention:

<dsname>(<all embedded dataset information>)

where <all embedded dataset information> will consist of the dataset name, followed by the word "EMB" within parenthesis if there is no set specification or followed by "EMB=<set name>" within parenthesis with a set specification; this will be repeated for all the levels of embedded structures. The maximum number of datasets that will be allowed is eight, due to a host file name limitation. An embedded dataset name(s) that is selected from the dataset list and added to the available table list will need to be "renamed" if the name is greater than 29 characters. A dialog box will appear which will indicate that the **table name length cannot exceed 29 characters**; you will need to rename the table, in the appropriate name box on this screen, to any desirable name that is less than or equal to 29 characters. A name that is provided for "rename" purposes will need to be different from any other name in the available table list; otherwise, a message will appear which will indicate that it is a duplicate table name. At this time, you will need to select "Cancel" or provide an appropriate name in the name box on this screen and click on "OK". An embedded dataset name in the available table list will have all parentheses replaced with hyphens. For query results, one row will be returned for each logical row; thereby, including the parent and embedded information.

An option to control the number of embedded levels that are displayed exists. To set this option, set the "value" attribute of the Open/A DMSIIDESC program. For example, WFL MODIFY "OBJECT/OPENA/DMSIIDESC/XX";VALUE=N where XX is the DMSII level and N is the number of levels.

Linked datasets, for **inquiry** purposes, may be selected and added to the available table list. The linked dataset(s) will appear in the dataset list with the following naming convention:

<dsname>(<linked dataset information>)

where <linked dataset information> will consist of the name of the dataset that is being linked to, followed by “LINK=<link name>“ within parenthesis. No sub-structure under a link will appear in the logical dataset reference.

All DMSII datasets, sets, and subsets, with the exception of manual subsets, are supported. Occurring links are not supported.

An example of the embedded dataset syntax is as follows:
DASDL representation:

```
DS1
  -DS2 embedded in DS1
     -DS3 embedded in DS2
     -DS5 embedded in DS2
        -DS6 embedded in DS5
  -DS4 embedded in DS1
     -DS8 embedded in DS4
  -DS7 embedded in DS1
```

In the dataset list, the embedded structure will appear as follows:

```
DS1(DS2(EMB=DS2-SET1) (DS3(EMB),(DS5(EMB)
(DS6(EMB))),DS4(EMB=DS4-SET1) (DS8(EMB)),DS7(EMB))
```

Hint: An “EMB” reference will refer to the preceding structure.

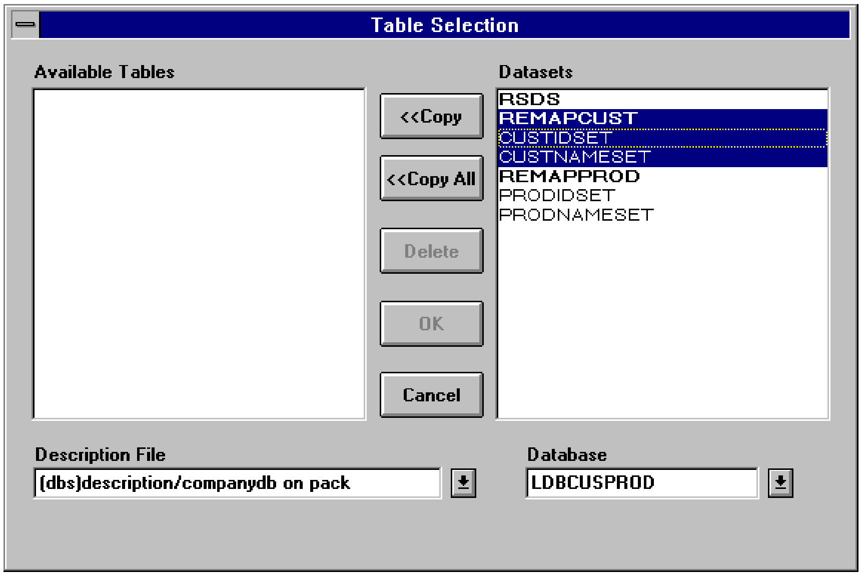
An example of the linked dataset syntax is as follows:
DASDL representation:

```
DS1
 -LUNPROTECTED linked to DS2
 -LCOUNTED linked to DS2
```

In the dataset list, the linked structure will appear as follows:

```
DS1(DS2(LINK=LUNPROTECTED),DS2(LINK=LCOUNTED))
```

NOTE: You must exit the Administration program before you may access an Open/A datasource from your workstation if it uses the same communications channel.



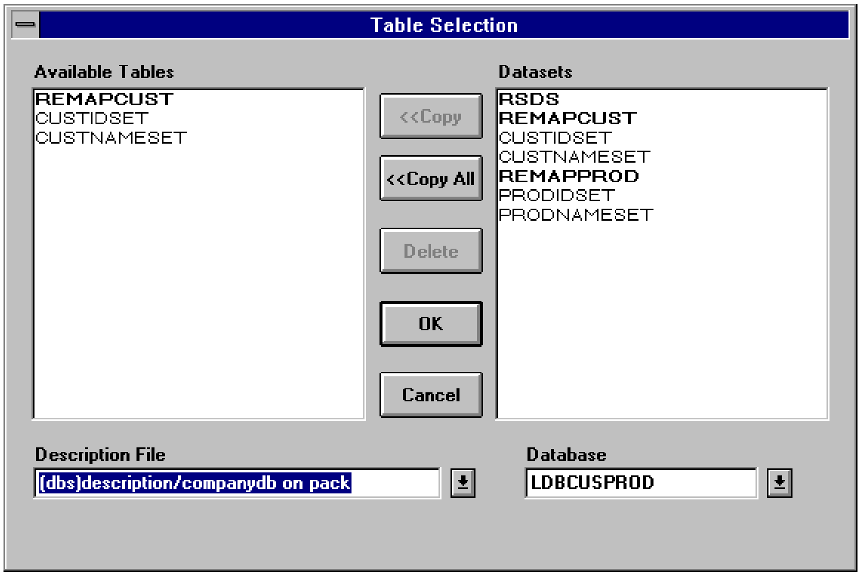
Datasets:

RSDS
REMAPCUST
CUSTIDSET
CUSTNAMESET
REMAPPROD
PRODIDSET
PRODNAMESET

Description File: [dbs]description/companydb on pack

Database: LDBCUSPROD

Dataset Selection



Available Tables:

REMAPCUST
CUSTIDSET
CUSTNAMESET

Datasets:

RSDS
REMAPCUST
CUSTIDSET
CUSTNAMESET
REMAPPROD
PRODIDSET
PRODNAMESET

Description File: [dbs]description/companydb on pack

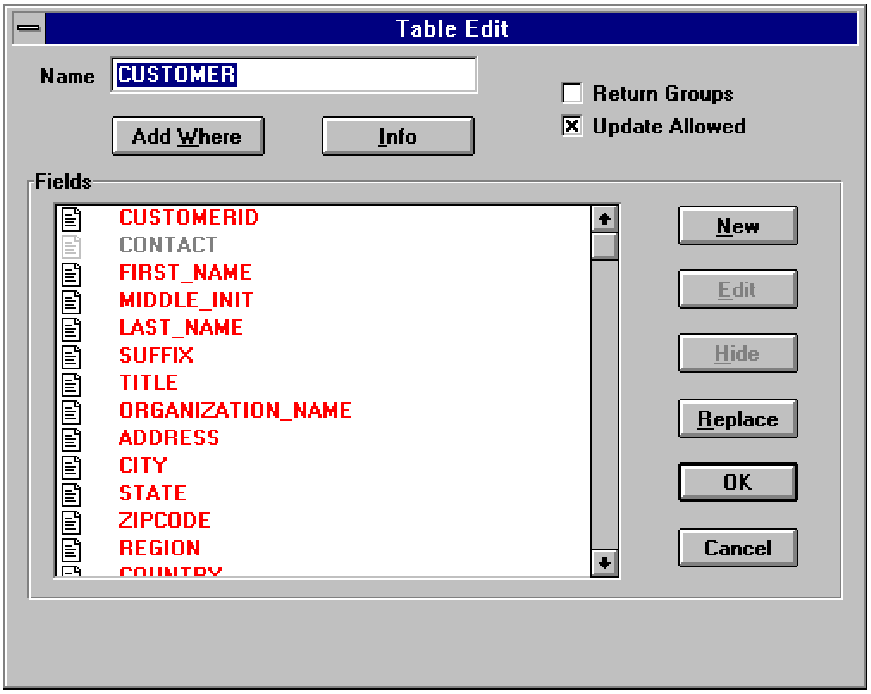
Database: LDBCUSPROD

Available Table List

Table Editing

Table names which appear in the table list on the Open/A window may be edited. You will be able to rename a table, specify whether or not group fields will be returned to the calling program, specify that the table is updateable, hide fields, rename fields, specify that the field(s) is updateable, specify the format type of the field(s), unhide the field(s), obtain the dataset name, set/subset name (if applicable), database name, and description file name, and use the wildcard replace feature where applicable.

In order to edit a structure, select it by clicking on the structure name, and then click on the "Edit" button or merely double click on the structure name. You will be presented with an "Edit Table" dialog box which will show you the table name, an update allowed box, a return groups box, an info box, a where condition box, and a fields box. The name, which may be changed, will be presented to any ODBC application as a table which may be accessed. A renamed table that contains a hyphen will appear in the Open/A table window with an underscore. In addition, the name that is used for rename purposes will need to be different from any other name in the available table list; otherwise, a message will appear which will indicate that it is a duplicate table name.



Name CUSTOMER

□ Return Groups
☒ Update Allowed

Fields

- CUSTOMERID
- CONTACT
- FIRST_NAME
- MIDDLE_INIT
- LAST_NAME
- SUFFIX
- TITLE
- ORGANIZATION_NAME
- ADDRESS
- CITY
- STATE
- ZIPCODE
- REGION
- COUNTRY

Table Editing Dialog Box

The **“Table Update Allowed” box** enables you to indicate that the table is updateable. Place an “x” in this box to indicate that this table is updateable; updateable tables appear in red in the Open/A table window.

The **“Return Groups” box** will allow you to specify whether or not group fields will be returned to the calling application.

The **“Info” box** provides table information relative to the actual dataset name, set/subset name (if applicable), database name, and description file name.

The **“Fields” box** provides a list of all the fields in the selected table. If the table has been marked as updateable, the field list will appear in red print; otherwise, if the table is not marked as updateable, the field list will appear in black print. Refer to the section titled “Field Editing” for information regarding the editing of fields.

The **“Add Where” box** allows you to specify a where condition for record selection on the table level. It will logically be AND’ED with any other criteria the user specifies; then, the entire select statement will be optimized.

In addition, you may also delete tables from the Open/A window. To delete tables from the list, select one or more tables and click on the “Delete” button in the Open/A table window.

Views

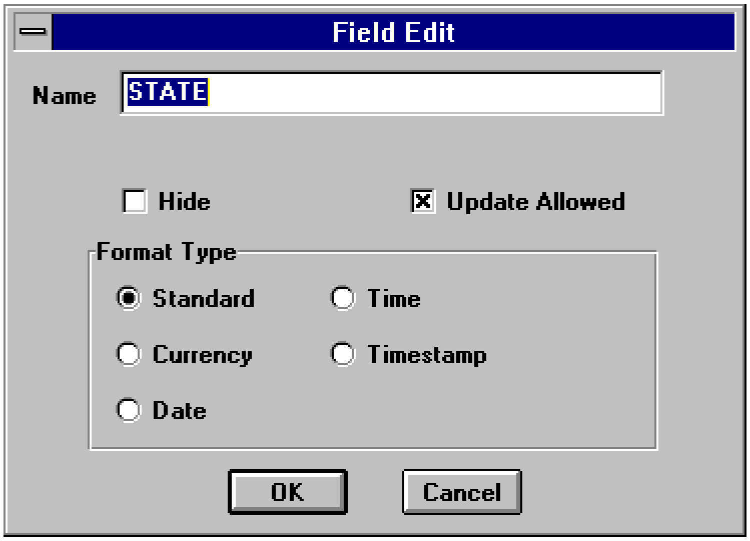
Views are created when changes are made to a table; ie: a field is hidden or renamed. Open/A automatically maintains the view files on the host machine; thus, users should **not** rename these files, etc. You will need to back up these files as necessary; refer to the “Backing Up Files” chapter. Note that the view files begin with a node of “OPENA”.

If no changes are made to a table, a view file will not exist.

Field Editing

Fields in a table may be renamed, hidden, specified as update allowed, declared as currency or date type, and user defined fields may be declared. These capabilities provide a site with more flexibility and control of their data. Fields are edited as indicated below.

1. From the “Edit Table” window, under “Fields”, select a field and click on the “Edit” field button.



Name: STATE

☐ Hide ☒ Update Allowed

Format Type

● Standard ○ Time
○ Currency ○ Timestamp
○ Date

Field Edit Dialog Box

2. To rename the field, enter the desired name in the “Name” box of the “Edit Field” dialog box. Hyphens in the name will be converted to underscores. The “rename” will need to be less than or equal to 29 characters.

You may also use the right mouse pop up menu to rename a field as documented in step 6 below.

3. To hide a field, place an “x” in the “Hide” box. A hidden field will appear as grayed out in the “Fields” list and will be invisible to an application.

If a field only needs to be declared as hidden, you may select the field and click on the “Hide” button; or, you may use the right mouse button pop up menu as documented in step 6.

You may “UnHide” a field that has been declared as hidden by selecting the field and clicking on the “UnHide” button; or, you may use the right mouse button pop up menu as documented in step 6.

4. A field may be specified as being updateable by placing an “x” in the “Update Allowed” box. You may also use the right mouse pop up menu to specify a field as updateable, as documented in step 6 below.

5. To specify a field as type currency, date, time, or timestamp, click on the appropriate button under “Format Type”. For fields of type date, a drop down box will appear for you to select the appropriate date format. For fields of type time, a drop down list will appear for you to select the appropriate time format. For fields of type timestamp, a drop down list will appear for you to select the appropriate timestamp format. The default format type is standard; basically a number or string.

You may also use the right mouse pop up menu to specify the format type of a field, as documented in step 6 below.

6. One or more fields may be edited by using the right mouse button pop up menu; select a field, or for multiple field selection, hold down the shift or ctrl key as you make your selections. By using the right mouse button pop up menu, you may hide the field(s), rename the field (if applicable), declare the field(s) updateable, and specify the format type.

A “Rename Field” dialog box appears when the “Rename” option is chosen via the right mouse button pop up menu. The “rename” will need to be less than or equal to 29 characters.

Once set, you may reset the hidden option and updateable option via the right mouse button pop up menu by re-selecting the option accordingly.

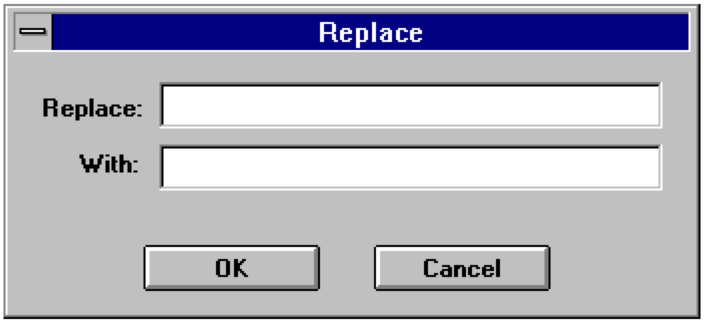
To change the format type using the right mouse button pop up menu, you need to select the applicable field format and then choose the appropriate format for the field from the drop down list.

7. You may select the "Cancel" option if you do not wish to set options; otherwise, select "OK".

Replace
The wildcard replace option allows you to find a text string and replace it with another string which you specify. This option applies to the entire field list.

In order to access the "Replace" field option, follow the steps below:

1. Click on the "Replace" button on the "Fields" section of the "Edit Table" window; a "Replace" dialog box will appear.



Replace Dialog Box

2. In the "Replace" location, enter the text which you wish to find.

3. In the "With" box, specity the text which you wish to be the replace text.

DATE, TIME, AND TIMESTAMP FIELDS

Open/A allows you to specify fields as containing date, time, or timestamp values. For example, if you have a field call "HIRE-DATE" that contains the date a person was hired in the format "MMDDYY", you may declare the field as "Date" with the type as "MMDDYY". The field will be analyzed and the calling application will receive the field back as a date. Time fields are similar to date fields except they contain time information(hour, minute, second). Timestamp fields contain both a date(year, month, day) and a time(hour, minute, second, fractional second). There are many available formats for date, time, and timestamp from which to choose in Open/A.

You may specify a date equality test in a "WHERE" condition, such as, <FIELD> > 'date literal'. There are three types of literals: date, time, and timestamp.

The date literal is of the format 'yyyy-mm-dd'. To perform a query and include all employees hired in 1995, you would use the condition "WHERE HIRE-DATE > '1995-01-01'.

The time literal if of the format 'hh:mm:ss' where hh is the hours(0-23), mm is the minutes(0-59) and ss is the seconds(0-59). For example, to specify 1:30PM, you would indicate '13:30:00'.

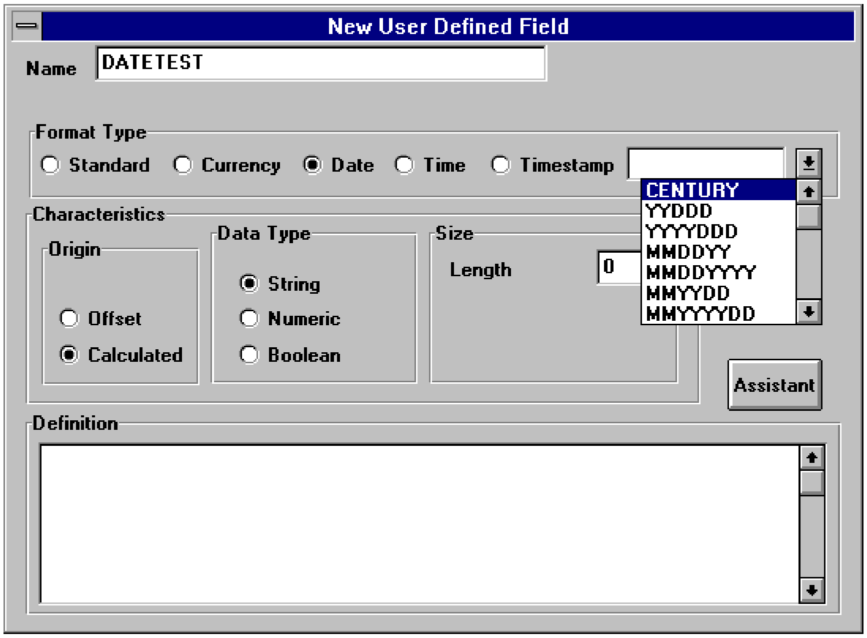
The timestamp literal is a combination of date and time. The format is 'yyyy-mm-dd hh:mm:ss[.ffffff]]'. Note the space between the date value and the time value. The following is a valid timestamp value:
'1995-08-01 11:34:02.1'

There are also several functions that deal with date, time, and timestamp.
For example, the following condition would include all orders placed in April: WHERE MONTH(ORDER-DATE)=4.

User Defined Fields

You may create a user defined field(s) which can be a calculation or a field that is offset in the data buffer. The format type, length, type of field, and definition will need to be specified. These user defined fields will appear in the “Fields” list of the “Edit Table” window and may be edited. To create a user defined field, follow the steps below.

1. Click on the “New” button on the “Fields” section of the “Edit Table” window.



New User Defined Field

Name DATETEST

Format Type

○ Standard ○ Currency ● Date ○ Time ○ Timestamp

Characteristics

Origin: ○ Offset ● Calculated

Data Type: ● String ○ Numeric ○ Boolean

Size: Length 0

Definition

New User Defined Calculated Field Dialog Box

2. On the “New User Defined Field” dialog box, enter a name for the field. The field name will need to be less than or equal to 29 characters. Any hyphens in the name will be converted to underscores.

3. Select the format type for this field; standard, which is the default; currency, date, time, or timestamp. For a date type field, a time type field, or a timestamp field, a drop down box will appear for you to select the appropriate format.

4. Under “Origin”, select the appropriate button for either a calculation or a field which will be offset in the data buffer.

The default is a calculated field. Calculated fields and offset fields are described below.

Calculation Field

A calculation is an expression. If the new field is declared as a calculation, you will be presented with a choice of string, numeric, or boolean for the data type. Depending on the data type selection for the field, you will be presented with additional dialog boxes as indicated below:

String

You will need to specify the length of the calculated field.
Same as an alpha field in DMSII

Numeric

You will need to specify the length of the calculated field and the number of decimal places that you desire.
Numeric values (integers with/without decimal places)

Boolean

Boolean items store information that has a logical value of either True or False. You only need to select the button for a boolean. 0 =False, 1 = True

Offset Field

An offset field is one that exists in the data buffer; its location in the data buffer is relative to some other field in the buffer; or, it may be declared at the start or end of the data buffer. The offset field may have a data type of alpha, packed, unpacked, real, boolean, or field. Depending on your data type selection, you will be presented with additional dialog boxes as indicated below:

Alpha

Alpha items are used to store character information. You will need to specify the length. Same as an alpha field in DMSII.

Packed, Unpacked

You will need to specify the length, the number of decimal places that you desire, and indicate if the field is signed. Packed is the same as a number field in DMSII; unpacked is number digits, but in character form

Real

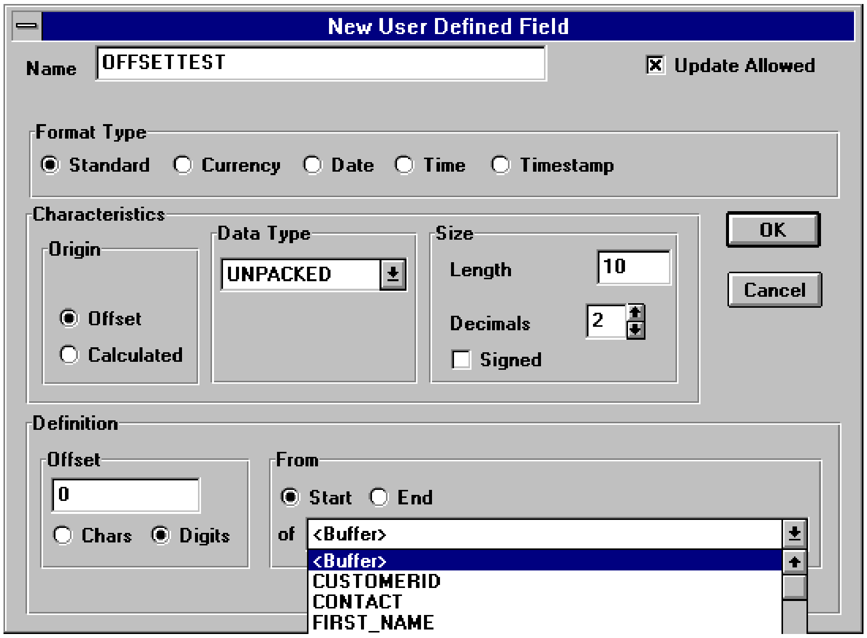
Real items are used to store signed or unsigned fractional or integer values. You will need to specify the length and the number of decimal places that you desire. Same as a DMSII real.

Boolean

Boolean items store information that has a logical value of either True or False. Each Boolean item occupies one 4-bit digit. For a boolean item, the buttons in the "Size" section of the window indicate which bit to reference in the actual data field. Since the rightmost bit contains information in a DMSII boolean, you will need to select the zero button, which is the default.

Field

Field items may be used to store Boolean values. You will need to specify the size of the item in bits; the item size must be an unsigned integer between 1 and 48. Individual bits will need to be named and declared as boolean to be used to store a logical value of True or False; True is represented by a value of 1 and False by 0.



New User Defined Field

Name OFFSETTEST

☒ Update Allowed

Format Type

● Standard ○ Currency ○ Date ○ Time ○ Timestamp

Characteristics

Origin

● Offset

○ Calculated

Data Type

UNPACKED

Size

Length 10

Decimals 2

□ Signed

Definition

Offset

0

○ Chars ● Digits

From

● Start ○ End

of <Buffer>

<Buffer>
CUSTOMERID
CONTACT
FIRST_NAME

New User Defined Offset Field Dialog Box

5. Specify the definition of the calculated field or the offset field in the definition part of the dialog box. The calculated field will need to be an expression; below are some examples.

Calculated Field Definitions:

FIRSTNAME & “ “ & LASTNAME
OR RTRIM(FIRSTNAME) & “ “ & LASTNAME

UNIT_PRICE * 1.25
QUANTITY * UNIT_PRICE
UNITS_IN_STOCK + UNITS_ON_ORDER

Offset Field Definitions:

Suppose zipcode is declared as 12 characters in the table design. You need to use the first five digits of the zipcode as a field and the next four digits of the zipcode as another field. You may use the “offset” field definition to redefine this zipcode field into two fields.

On the New User Defined Field dialog box, enter the following definition in order to define the first five digits of the zipcode:
“ZIP5” for a name
“Standard” for the format type
“Offset” for the origin
“Alpha” as the data type
“5” as the length
“0” characters from the start of the field “ZIPCODE”

For the next four digits of the zipcode field, enter the following definition for another user defined field:
“ZIP+4” for a name
“Standard” as the format type
“Offset” as the origin
“Alpha” as the data type
“4” as the length
“5” characters from the start of the field “ZIPCODE”

6. Click on “OK” ; the new field will appear in the field list.
   You may click on “Cancel” if you do not wish to create a new
   user defined field at this time.

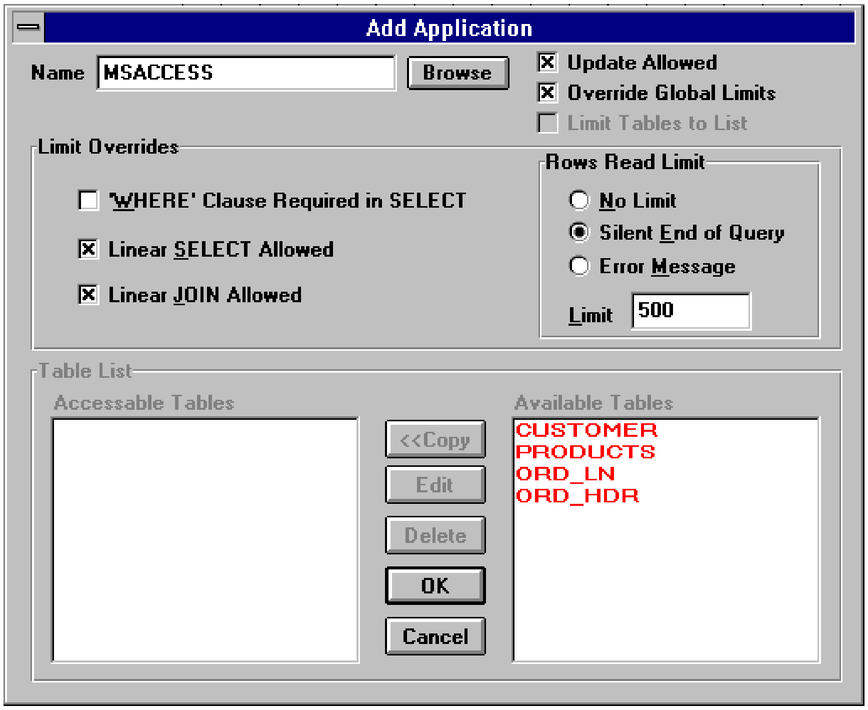
Application Editing

PC application programs may be defined as having update capability to the available tables. New application programs may be defined as having update capability and existing application programs may be edited. Also global limits may be overriden on the application level.

Since an application program, by default, may inquire on available tables, there is no need to define an application program which will only have inquiry access to available tables. However, with a future option, “Limit to List”, you may be able to limit which of the available tables that this application may have inquiry or update access to. In addition, you may wish to override the global limits for this application program.

An application will be able to update a table if the table has been marked as updateable and the application has been marked as “update allowed”. Otherwise, the application will only be able to inquire on the table.

The application dialog box is presented when you click on “Apps” and then click on “New”; or, for application editing, click on the appropriate application name and then click on “Edit”.



Add Application

Name MSACCESS

☒ Update Allowed

☒ Override Global Limits

☐ Limit Tables to List

Limit Overrides

☐ 'WHERE' Clause Required in SELECT

☒ Linear SELECT Allowed

☒ Linear JOIN Allowed

Rows Read Limit

○ No Limit

● Silent End of Query

○ Error Message

Limit 500

Table List

Accessable Tables

Available Tables

CUSTOMER
PRODUCTS
ORD_LN
ORD_HDR

PC Applications Dialog Box

An application may be deleted from the application list on the Open/A window by selecting the desired application name and clicking on “Delete”.

Open/A supports the usage of asynchronous processing. If your application supports asynchronous processing, you may use this feature. To set this option in Open/A, put the following line of text in your “opena.ini” file, located in the Windows directory, in the [ODBC] section:

```
allow_async=1
```

Global Limits

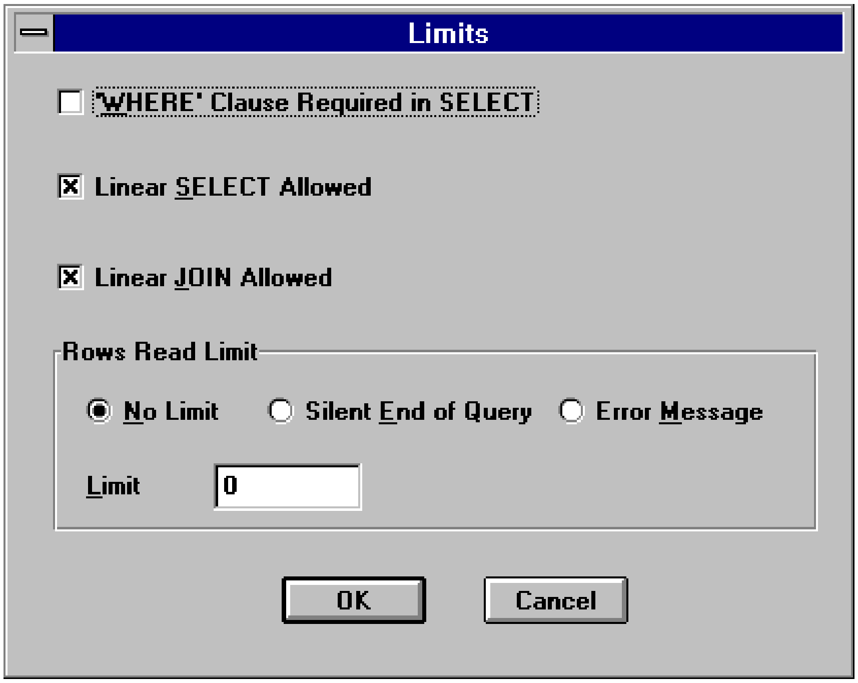
On the Open/A table window, there is a “Limits” button. This option will allow you to globally specify whether the “WHERE” clause is required in select statements, allows you to specify if a linear search is allowed, allows you to specify if a linear join is allowed, and allows you to specify a rows read limit.

The rows read limit will allow you to indicate if there is no limit in the number of rows to be read, or if a message will be given if there are additional rows to be read after reading the specified “limit” rows, or if no message will be given if there are additional rows to be read after reading the specified “limit” rows.

To access the Limits option, follow the steps below:

1. Click on the “Limits” button on the Open/A table window.

2. Click on the “Edit” button on the Open/A table window.



Limits

□ 'WHERE' Clause Required in SELECT

☒ Linear SELECT Allowed

☒ Linear JOIN Allowed

Rows Read Limit

● No Limit ○ Silent End of Query ○ Error Message

Limit 0

Global Limits Dialog Box

3. Put an “x” in the check box(es) applicable to the option which you wish to set.

4. Click on the option button “No Limit” if you wish to indicate no read limits, relative to the number of rows to read.

5. Click on the option button “Silent End of Query” to indicate that only the number of rows as specified in the Limit box will be read. No message will be given to indicate additional rows exist in the file.

6. Click on the option button “Error Message” to indicate that only the number of rows as specified in the Limit box will be read. In addition, a message will be given to indicate that additional rows exist in the file.

Global Client Configuration File Settings

A client configuration file may have its own global or custom settings. These settings are defined on a custom client configuration file dialog box. **To access the global or custom client configuration file settings, click on "Global" on the Open/A table window; then, click on "Edit".**

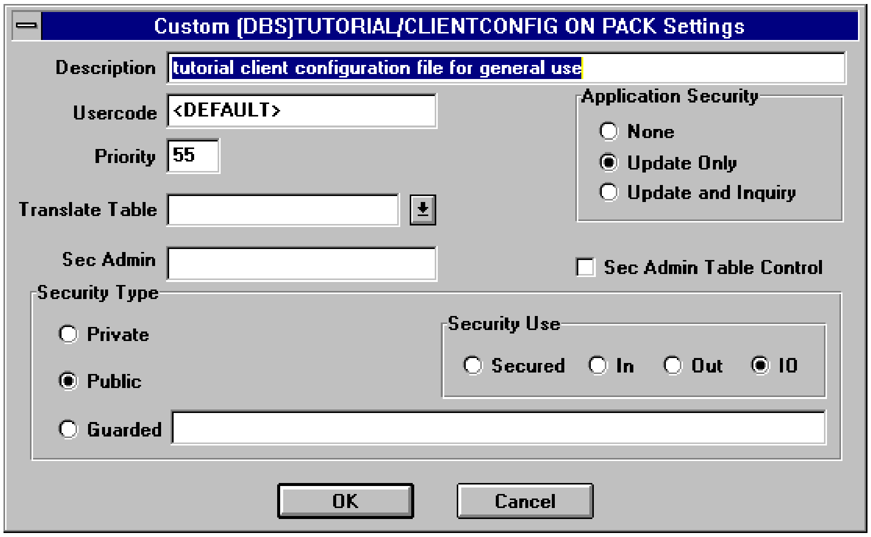
A general description field exists for any descriptive text which you wish to enter for a client configuration file. **The usercode setting represents the usercode under which you wish the user to run when using this client configuration file. It may be a different usercode from the one that is specified in the datasource that is used to log the user onto the host.** The default usercode is the one under which the user logged onto the host.

If a priority is specified, the user's session will run with that priority; otherwise, the default is the priority of the Open/A program.

The application security may be no security, update only, or update and inquiry. No security indicates that any PC application can inquire and/or update; Update Only indicates that any PC application can inquire; however, an application must be in the list and marked as "update allowed" in order to update; Update and Inquiry indicates full security whereby PC applications may have update and inquiry capability; however, the PC applications will need to be in the list in order to inquire or update.

You may set the option "Use Database Libraries" if you wish to have one database library per user. When set, all client updates for a database will be performed in the same transaction.

The client configuration file security type will need to be specified; the default is public. The security type may be public, private, or guarded with a guard file name specified. The security use of the client configuration file may be SECURED, IN, OUT, or IO; the default is IO.



Custom (DBS)TUTORIAL/CLIENTCONFIG ON PACK Settings

Description: tutorial client configuration file for general use

Usercode: <DEFAULT>

Priority: 55

Translate Table:

Sec Admin:

Application Security: ○ None ● Update Only ○ Update and Inquiry

□ Sec Admin Table Control

Security Type: ○ Private ● Public ○ Guarded

Security Use: ○ Secured ○ In ○ Out ● IO

Custom Client Configuration File Settings

Chapter 4

Installation and Use of Open/A HLCN Connection

Overview

If you are running Unisys' HLCN software on your A-Series and have NETBIOS software installed on your PC, you may use the Open/A direct HLCN connection. The advantages to this method are performance and ease of installation. You are not required to have a dedicated Open/A station on each PC, you do not have to have a terminal emulator, and you do not have the added overhead of going through COMS and the direct window program. The sections that follow describe the steps that must be performed on the A-Series and the Windows PC in order to enable Open/A to communicate via HLCN.

Mainframe Setup

There is a program that is installed during the Open/A A-Series installation process which must be running in order for the direct HLCN connection to be available. This program will be on the same usercode and pack as the remainder of the Open/A files; it is titled OBJECT/OPENA/COMM/HLCN. In order for the HLCN connection to always be available, this program will need to be RUN from the Supervisor, a restartable WFL, or your operations staff should be instructed to RUN it after the system has been brought up. Once this program is running, the Open/A Windows clients will be able to connect to the A-Series and access all of the Open/A functionality. By default, 128 active concurrent connections will be allowed by the HLCN handler. If you would like to allow a different number, you will need to set the "TASKVALUE" task attribute to the desired number. This may be done in the RUN statement or by a WFL MODIFY statement to modify the TASKVALUE attribute. For example, the following command would run the HLCN handler and allow 256 active connections

```
RUN (OPENA)OBJECT/OPENA/COMM/HLCN ON PACK;VALUE=256
```

There are several commands which may be sent to the running Open/A HLCN program. These commands should be sent to the mix number of the Open/A HLCN program via the A-Series "AX" command. The commands and their actions are listed below:

1. **STATUS** - will display the number of allowed and currently active connections
2. **WRU** - will display the version of the HLCN program
3. **QUIT** - will shutdown the HLCN program (this will only work if there are no active connections)
4. **ABORT** - same as QUIT but works even if there are active connections

An option is available that will shut down a connection handler in situations whereby a client workstation is rebooted and the connection handler for that station is still running. This option will need to exist in the OPENA/CONFIG file, located on the same usercode and pack as the Open/A files, with the following syntax:

HLCNCONNTIMEOUT=XX

where XX is the time interval, in number of minutes, that the connection will be checked for the absence of the client workstation. For example, if you are running TCP/IP and you wish to have the connections checked every 5 minutes, the following line would need to appear in the OPENA/CONFIG file:

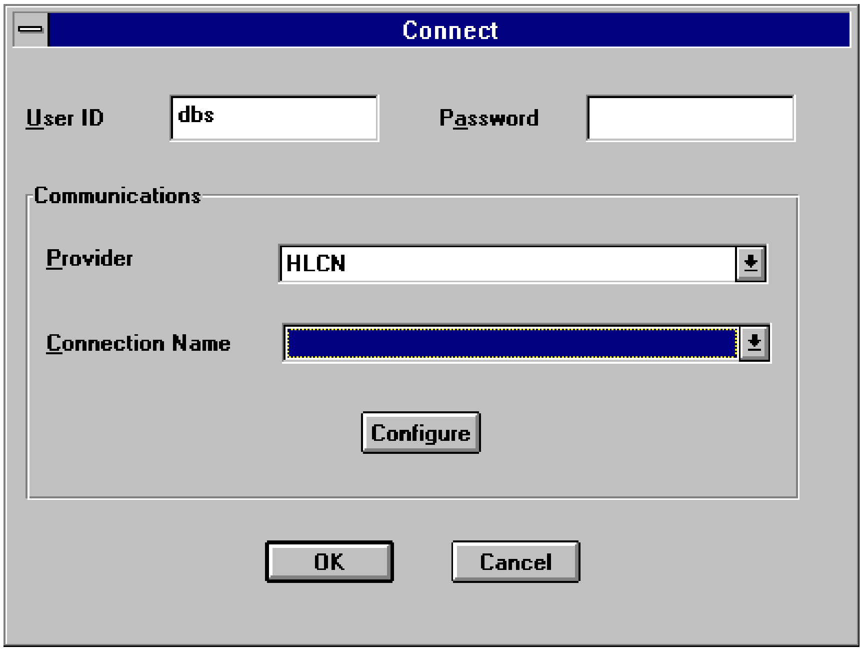
HLCNCONNTIMEOUT=5

Windows PC Setup

In order to use the Open/A direct HLCN connection, you must have a NETBIOS driver loaded on your PC. Once you have NETBIOS software installed and functioning, you will see the choice “HLCN” in the Open/A Administration connection dialog and in the Open/A ODBC datasource setup.

The following setup may be done in either the Open/A Administration connect dialog box or from the Open/A ODBC datasource configuration.

1. Select the HLCN Communications Provider. Notice that the “Configure” button should turn from gray to black.



User ID: dbs

Provider: HLCN

Figure 17a - Open/A Connect Dialog Box with HLCN Provider

2. Click on the “Configure” button. Once you select this button, you will be presented with the “HLCN Transport Setup” dialog box.

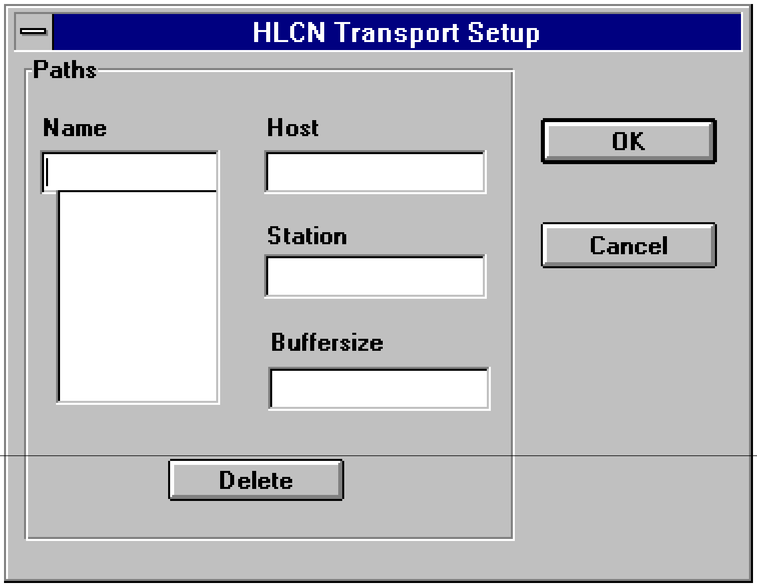


Figure 17b - HLCN Transport Dialog Box

3. In the transport setup dialog “Name” box, enter a descriptive name for the connection to your host. It may be the same name as the host or some other name. For example, if you are configuring a connection to your host called “PRODA11”, you may want the connection name to be “PRODA11_PATH”.
4. Enter the actual A-Series host name, as shown in the WM system command, in the “Host” box.
5. In the “Station” box, enter a unique descriptive “station name” for your connection. This station name will be used to identify the client on the A-Series and should follow A-Series station naming conventions.
6. You may specify a value in the buffersize location in the dialog box, if desired. This value represents the size of the buffer which is used to send a message. If the buffersize is set on the PC, it will need to be specified on the host in the OPENA/CONFIG file, located on the same usercode and pack as the Open/A files as follows:

   ```
   HLCNBUFFERSIZE=XXXX
   ```

   Open/A will use the larger of the host and PC buffersize values. In addition, each PC may have a different buffersize value. The value may not be less than 1920 characters and may not be greater than 8192 characters.

7. Click “OK” once you have entered the host and station name.

If you now click on the drop down box for “Connection Name”, you will see the name that was entered previously in the transport setup dialog box. In our example, the name is “PRODA11_PATH”. After selecting this name, you are ready to connect to the A-Series.

NOTE: You may also delete a connection. Select the name of the connection which you wish to delete from the “Name” list of the transport dialog box. Click on the “Delete” button on the transport setup dialog box.

Chapter 5

Installation and Use of Open/A TCP/IP Connection

Overview

If you are running Unisys TCP/IP software on your A-Series and have TCP/IP software installed on your PC, you may use the Open/A direct TCP/IP connection. The advantages to this method are performance and ease of installation. You are not required to have a dedicated Open/A station on each PC, you do not have to have a terminal emulator, and you do not have the added overhead of going through COMS and the direct window program. The sections that follow describe the steps that must be performed on the A-Series and the Windows PC in order to enable Open/A to communicate via TCP/IP.

Mainframe Setup

There is a program, that is installed during the Open/A A-Series installation process, which must be running in order for the direct TCP/IP connection to be available. This program will be on the same usercode and pack as the remainder of the Open/A files. The name of the program is OBJECT/OPENA/COMM/TCPIP. In order for the TCP/IP connection to always be available, this program should be RUN from the Supervisor, a restartable WFL, or your operations staff should be instructed to RUN it after the system has been brought up. Once this program is running, the Open/A Windows clients will be able to connect to the A-Series and access all of the Open/A functionality. By default, 128 active concurrent connections will be allowed by the TCP/IP handler. If you would like to allow a different number, you will need to set the “TASKVALUE” task attribute to the desired number. This may be done in the RUN statement or by using the WFL MODIFY statement to modify the TASKVALUE attribute. For example, the following command would run the TCP/IP handler and allow 256 active connections:

```
 RUN (OPENA)OBJECT/OPENA/COMM/TCPIP ON PACK;VALUE=256
```

There are several commands which may be sent to the running Open/A TCP/IP program. These commands should be sent to the Open/A TCP/IP mixnumber via the A-Series “AX” command. The commands and their actions are listed below:

1. STATUS - will display the number of allowed and currently active connections
2. WRU - will display the version of the TCP/IP program

3. QUIT - will shutdown the TCP/IP program (this will only work if there are no active connections)
4. ABORT - same as QUIT but works even if there are active connections

An option is available that will shut down a connection handler in situations whereby a client workstation is rebooted and the connection handler for that station is still running. This option will need to exist in the OPENA/CONFIG file, located on the host under the OPENA usercode, with the following syntax:

TCPIPCONNTIMEOUT=XX

where XX is the time interval, in number of minutes, that the connection will be checked for the absence of the client workstation. For example, if you are running TCP/IP and you wish to have the connections checked every 5 minutes, the following line would need to appear in the OPENA/CONFIG file:

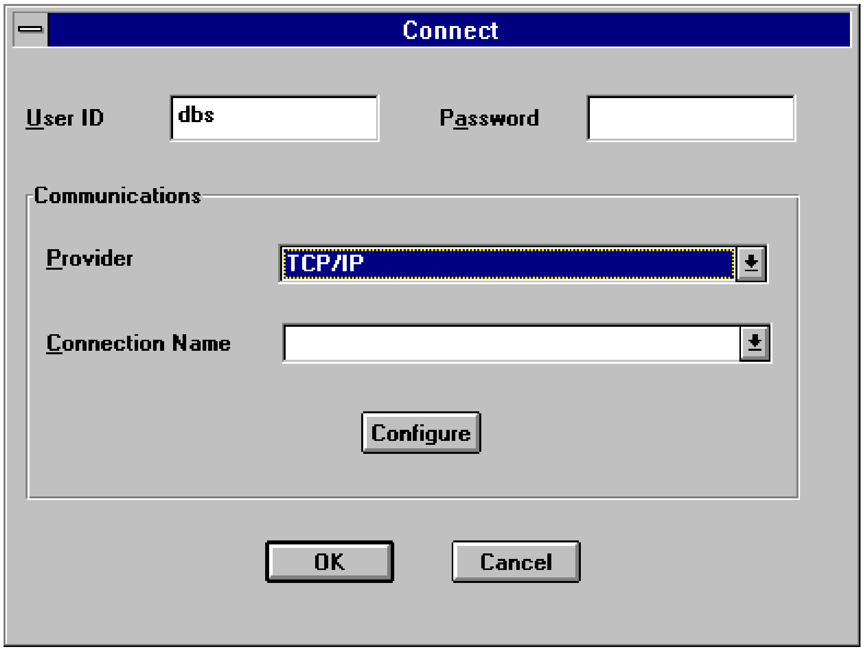
TCPIPCONNTIMEOUT=5

Windows PC Setup

In order to use the Open/A direct TCP/IP connection, you must have a TCP/IP product installed on your PC which supports the WINSOCK standard. Once you have the TCP/IP software installed and functioning, you will see the choice “TCP/IP” in the Open/A Administration connection dialog and in the Open/A ODBC datasource setup. If this choice is not shown and you have the TCP/IP software installed correctly, the problem is probably that the WINSOCK library, WINSOCK.DLL, is in a directory that is not in your DOS PATH. If this is your problem, you should either add this directory to your path or move the WINSOCK.DLL to the Windows system directory.

The following setup may be done in either the Open/A Administration connect dialog box or from the Open/A ODBC datasource configuration.

1. Select the TCP/IP Communications Provider. Once TCP/IP is selected, the “Configure” button should turn from gray to black.



Connect

User ID: dbs

Password

Communications

Provider: TCP/IP

Connection Name

Figure 17c - Open/A Connect Dialog Box

2. Click on the “Configure” button. You will be presented with the “TCP/IP Winsock Transport Setup” dialog box.

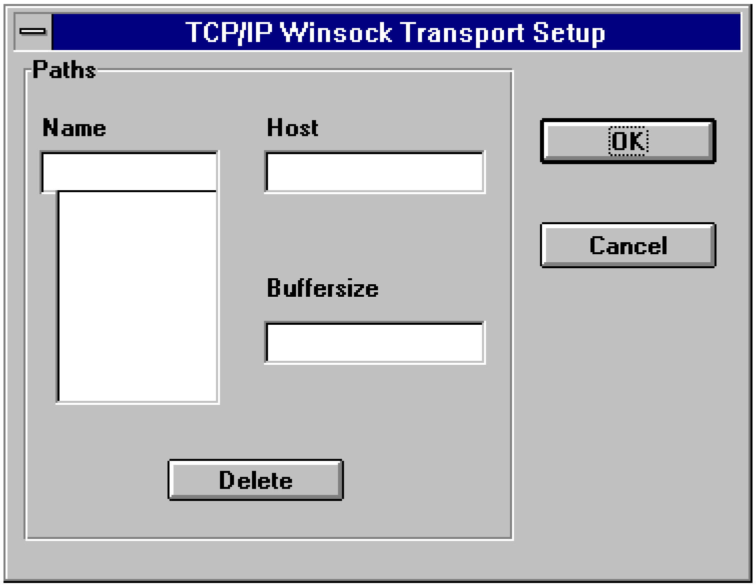


Figure 17d - TCP/IP Winsock Transport Dialog Box

3. In the transport setup dialog "Name" box, enter a descriptive name for the connection to your host. It may be the same name as the host or some other name. For example, if you were configuring a connection to your host called "PRODA11", you might want the connection name to be "PRODA11_PATH".

4. In the "Host" box, enter the actual TCP/IP host name for your host. If you do not have a "HOSTS" file setup on your machine, you may enter the actual IP address of the host.

5. You may specify a value in the buffersize location in the dialog box, if desired. This value represents the size of the buffer which is used to send a message. If the buffersize is set on the PC, it will need to be specified on the host in the OPENA/CONFIG file, located on the same usercode and pack as the Open/A files as follows:

   TCPIPBUFFERSIZE=XXXX

   Open/A will use the larger of the host and PC buffersize values. In addition, each PC may have a different buffersize value. The value may not be less than 1920 characters and may not be greater than 8192 characters.

6. Click on “OK” after you have entered the host name.

If you now click on the drop down box for “Connection Name”, you will see the name that was entered previously in the transport setup dialog box. In our example, this name is “PRODA11_PATH”. After selecting this name, you are ready to connect to the A-Series.

NOTE: You may also delete a connection. Select the name of the connection which you wish to delete from the “Name” list of the transport dialog box. Click on the “Delete” button on the Transport Setup Dialog box.

Chapter 6

Setting Up COMS Direct Window Program And Stations

Introduction

Currently, the installation utility does not set up the necessary COMS direct window, program and agenda. You must use COMS Utility in order to set these entities up manually. The installation process is described below.

Setting Up COMS Direct Window Program

NOTE: Follow the steps below if you are not using direct HLCN or if you are not using direct TCP/IP.

You are now ready to set up the COMS direct window program. This is the program that handles the communications between the Windows based PC and the Open/A mainframe software. If the files on the A-Series distribution tape were copied to usercode "OPENA" on the diskpack named "PACK", then the codefile name of the communications handler will be (OPENA)OBJECT/OPENA/COMM/DIRECT ON PACK. You may run any number of copies with any number of max users that you wish. For those of you familiar with COMS, you may set up the program and skip the remainder of this section.

For those of you less familiar with COMS, the following is an example of the pertinent COMS utility screens, relative to MCP 42.3 release.

The first screen is the program definition:

```
P - PROGRAM ACTIVITY                                   COMS
 Action: CR
        CReate  MOdify  INquire  DElete  GO  Home  (Press SPCFY for
help)
        SEarch  FIrst   LAst     NExt    PRevious  DUmp

Program Name  . . . . . . . . . OPENA
Task Attributes
  TITLE  . . . . . . . . . . . (OPENA)OBJECT/OPENA/COMM/DIRECT ON PACK

  USERCODE . . . . . . . . . ..      .
  ACCESSCODE                         .
.   CHARGECODE . . . . . . . . . .
  ( Additional Task Attributes Can Be Specified On The PAT Screen )
Valid Security-Category List    ALL
Database Name                  NONE
Input-Queue Memory Size (words) 0       Init Time Limit     0  : 0
Minimum Copies                 1        Term Time Limit 0  : 0
Maximum Copies  . . . . . . . . 1        Queue Depth . .  0
Remote-File Interface  (Y/N)     N       Remote Users    0
Have COMS Perform TXOPEN (Y/N) N
Installation Data Name          NONE
```

The screen defining the window is shown below:

```
W - WINDOW ACTIVITY (DIRECT)                         COMS
 Action: CR
        CReate  MOdify  INquire  DElete  GO  HOme  (Press SPCFY for help)
        SEarch  FIrst   LAst    NExt    PRevious  DUmp

 Window Name. . . . . . . . OPENA
 Subaddress
 Window Type. . . . . . . . D              D - Direct  R - Remote-File  M -
MCS
 Virtual Terminal Name    DEFAULT
 Maximum Users. . . . . . . 10              Maximum Dialogs. . . . . . 5
 Notify Open (Y/N) and Text N
 Notify On (Y/N) and Text .   N
 Installation Data Name    NONE
 Direct Window specific fields:
   Window Trancode Parsing  NONE
   Maximum Trancode Size. . 0
   Protection (OUTPUT/NONE)  NONE
      Protected Output File
```

And finally, the agenda screen is shown below:

```
A - AGENDA ACTIVITY                                  COMS
 Action: CR
          CReate  MOdify  INquire  DElete  GO  HOme   (Press SPCFY for help)
          SEarch  FIrst   LAst     NExt    PRevious  DUmp

Agenda Name . . . . . . . . . . OPENAINPUT
Window Name                     OPENA
Default Input Agenda (Y/N)  . .      Y
Default Output Agenda (Y/N)          N
Transaction-Mode Agenda (Y/N) . N

Processing Items:
   Processing-Item List . . . . NONE
   Destination                   OPENA

Input Queue Protection (Y/N)    N      Wait For Completion (Y/N) N

Set Transaction Mode (Y/N)  . . N
Installation Data Name          NONE
```

After setting up the direct window program, you may test it from any station. Just enter a **?ON OPENA**. Then, type any text and transmit. You should receive a message stating that this station is attached to the Open/A mainframe software.

Setting Up Stations

1. Modify the configuration of any stations that you are going to use with the Open/A software. Each station will need to be given a default user code and continuous login. In addition, the default window for each of these stations should be OpenA. An example of the COMS utility screen for such a station is shown below.

```
 S - STATION ACTIVITY                              COMS
 Action: IN
     CReate  MOdify  INquire  DElete  GO  HOme help (Press SPCFY for
help)
     SEarch  FIrst   LAst     NExt    PRevious  DUmp
 Station Name . . . . . . . . DCDLP_TDSTA12


                                        ]
 Hostname . . . . . . . . . .  SHARON
 Device Type                   DEFAULTDEVICE
 Defaults:
     Usercode     OPENA                   Window     OPENA
     Accesscode NONE                      Trancode    NONE
     Chargecode NONE
 Valid Security-Category List  ALL
 Trancode Position  . . . . .  1       Trancode Override Allowed (Y/N)  N
 Control Station (Y/N)         N       Privileged User (Y/N)              N
 Super User (Y/N) . . . . . .  N       System User (Y/N)  . . . . . . .   N
 Continuous Log-on (Y/N)  Y            Timeout Interval         0: 0
 Close Action . . . . . . . .   1      Close Window . .  MARC
     1) Close Window  2) Previous  3) No Window  4) None Specified
 Installation Data Name       NONE
```

Only the “Default Window”, “Default Usercode” and “Continuous Log-on” must be set as indicated. All other values may be set to any value that your site requires. You may assign any non-privileged usercode with minimal access to the A-Series. Its sole purpose is to enable the communications link with the Open/A software. Each time a client application connects to the A-Series, the appropriate usercode will be assigned by the Open/A direct window program after the usercode and password of the user have been validated.

2. After setting up the station, you will need to go into a terminal emulator and make sure that transmitting text will cause you to receive the message about the station being attached to the Open/A software.

NOTE: If you are using INFOConnect with HLCN or LCW, the workstation default window will override the value set in COMS Utility. If you are running in this environment, you will need to set the default window in the INFOConnect path to “OPENA”.

Scripting

NOTE: Do not use this feature if you are using direct TCP/IP or direct HLCN.

Scripting allows a site to access Open/A through a station that does not have continuous log-on set and/or does not have Open/A as the default window. **However, it is recommended that continuous log-on be set and that Open/A be the default window.** Scripting is enabled via two files; a script file located in any directory on the PC (we recommend that it reside in the OpenA directory); and by the opena.ini file, located in the Windows directory on the PC. Refer to the information below for details on the format of the script file and for information on the required section in the opena.ini file.

Script File Format

The name of this file must be the name specified in the opena.ini file. The script file will contain a Connect section and a Disconnect section; the "connect" section is executed when Open/A connects to the A-Series, the "disconnect" section is executed when Open/A disconnects from the A-Series. Within each section, the commands, "timeout", "receive", and "send" are applicable. These commands are defined below. The script file is a **text** file and will need to be created via "Notepad", or similar PC application:

The commands are as follows:

timeout <n>

-set receive timeout to "n" seconds where n is between 1 and 254, inclusive.

receive <text>

-if <text> is specified, the Open/A script processor receives messages until one containing <text> is received or a timeout occurs. if <text> is not specified, the Open/A script processor waits for the next message or a timeout.

NOTE: once a timeout occurs, all "receives" immediately following, until a timeout or send command, will be skipped.

send "<text>"

-sends the quoted text to the host. The special sequences "#userid" and "#password" will be replaced with the actual usercode and password specified in the administration connect screen or ODBC datasource.

An example of a script file follows:

```
[CONNECT]
send    "#userid  #password"
timeout 2
receive
receive
receive
send"?ON OPENA"
receive"OPENA"
[disconnect]
timeout 2
send"?close purge"
receive
receive
send "?on marc"
receive
receive
send "bye purge"
```

If Open/A cannot connect to the host, you will need to set trace in order to debug the script file. This will enable you to verify that the script file is correct. Refer to the information below regarding how to set the trace command within the "opena.ini" file to enable debugging of the script file.

opena.ini

In the opena.ini file, located in the Windows directory, you will need to have a script section declared, as described below, in order for scripting to function. This file is a **text** file; if this file does not already exist in your Windows directory, you will need to create it via "Notepad", or a similar PC application.

```
[script]
trace=0
tracefile=<complete text file name, including path>
<provider>_<station name>_scriptfile=<path>:\<script file name>
```

The commands are discussed below:

To set trace to enable debugging, you will need to replace the zero with a 1 in the line that reads "trace=0". To reset tracing in order to turn debugging off, you will need to replace the 1 with a zero.

The **tracefile** will need to refer to the name of the trace file that will be created if the trace option is set.

The last command above refers to the script file. In this command, you will need to reference the provider and the station name. The <provider> will be INFOCONNECT or CTCBRIDGE. The <station name> will be your path name if you are using INFOConnect or it will be your station name if you are using CTCBridge.

Open/A will use the <provider>_<station name> reference, if it appears, to determine the name of the script file, if any, to use for the specified station. If the <provider>_<station name> reference does not appear, Open/A will look to see if the following reference exists:

<provider>_scriptfile=<path>:\<scriptfile name>

This reference will allow you to globally specify a script file name for stations running under a given provider.

Chapter 7

Transport Specifics

Upstanding NTP

Upstanding Systems requires a driver for usage with their Upstanding NTP transport. You may obtain this driver by calling Upstanding Systems. You will need to specify the directory of the Upstanding software in your path statement.

Upstanding does not allow a station to be reused; thus, if you wish to allow a station to reuse its address automatically instead of having to manually clear the station, follow the steps below:

1. Get the sample source program
   "UPSTANDING/EXAMPLE/UERSERVICES", located on the standard distribution tape.
2. At the top of the file, add the following line of text:
   "$SET DUPLICATE_NAME_RESOLVER"
3. Line 20838000 should read "ACTION[RESULT]:=DEFAULTV;"
   Change this line to read "ACTION[RESULT]:=USE_EXISTINGV;"
4. Perform the same change described in step 3 to the line 21136000
5. Compile the source and save the object file
6. Enter a command at the ODT of the form:
   SL UPUSERSERVICES=<object name> where <object name> is the name of the object file that was saved in Step 5

After performing all of these steps, you will no longer need to manually clear any Open/A stations.

CTCBRIDGE

If you are not using poll select “ctcdos.dll” with CTCBRIDGE, you will need to put the following reference in the opena.ini file, located in your Windows directory:

```
[ctcbridge]
driver=<name>
```

where <*name*> is the name of the specific driver you are using
ie: driver=ctcipx.dll

Secondly, if the CTCBRIDGE software is not installed in the “c:\ctcbr” directory, you will need to put the following reference in the opena.ini file, located in your Windows directory:

```
[ctcbridge]
ctcpath=<x>:\<directory>
```

where <*x*> is the drive and <*directory*> is the directory
ie: ctcpath=d:\ctcbr

TELNET

Telnet uses Winsock to talk to the COMS Open/A window for Unigate, Unisys and Upstanding TCP/IP.

You will need to provide the host name and a connection name.

APIS

If OCTUS is not in the "c:\octus" directory, you will need to put the following reference in the opena.ini file, located in the Windows directory:

```
[apis]
path=<x>:\<directory>
```

where <*x*> is the drive and <*directory*> is the directory
ie: path=d:\testapis

HFS

You will need to set up the COMS direct window interface. You will need to provide a connection name and indicate which of the terminal numbers you wish HFS to work with.

Chapter 8

Joins in the Open/A Environment

Inner and Outer Joins

INNER and OUTER joins are supported.

INNER JOIN Selects records from the primary table where there is a match on the corresponding join field(s) in the secondary table.

OUTER JOIN Selects all records from the primary table and only those records from the other table where there is a match on the joined field(s). A NULL will be returned for all fields in the secondary table if there is no match on the join field(s).

Inner and Outer Join Examples

INNER JOIN:

I.E.: SELECT SUPPLIERS.COMPANY_NAME, SUPPLIERS.SUPPLIER_ID, PRODUCTS.PRODUCT_ID, PRODUCTS.PRODUCT_NAME FROM SUPPLIERS, PRODUCTS WHERE SUPPLIERS.SUPPLIER_ID = PRODUCTS.SUPPLIER_ID

This join will return all records from the Suppliers table which have a match the supplier id field in the Products table. Data returned will be the company and supplier id from the Suppliers table, and the product id and name from the Products table.

OUTER JOIN:

SYNTAX: {oj <table1> LEFT OUTER JOIN <table2> ON <condition>}

NOTE: table1 may be an outer join

I.E.:

```
SELECT CUSTOMERS.COMPANY_NAME,
ORDERS.ORDER_ID, ORDERS.ORDER_DATE FROM
{OJ CUSTOMERS LEFT OUTER JOIN ORDERS ON
CUSTOMERS.CUSTOMER_ID =ORDERS.CUSTOMER_ID}
```

This join will return all records from the Customers table and only those records from the Orders table which match the customer id field in the Customers table. Data returned will be the company name from the Customers table, order id and order date from the Orders table.

Chapter 9

KEYEDIOII Files and FLAT Files

KEYEDIOII and FLAT FILES

Open/A supports the usage of KEYEDIOII files and flat files in either query or update mode. The administration program provides a way of defining the record layout for these file types since a description file does not exist for them. You may manually construct a flat description file and a data file layout that describes your data file as documented below; or, you may use the Cobol parser functionality which will allow you to specify that new tables are to be created from a COBOL source program; thereby **eliminating** the need to manually construct a flat description file and a data file layout for flat files and KIOII files. Both functionalities are supported.

COBOL PARSER

You may set an option which will specify that tables are to be created from a COBOL source(may be used for KIOII and Flat Files) . To set this option, follow the steps below:

1. Select “Special” on the menu bar.

2. Choose “Preferences” from the drop down list; click in the box which reads “New tables from COBOL Source”.

This option will remain in effect until it is turned off by clicking in the selected box again.

3. Once a configuration file is selected, you may select “Table”, click on “New”, and the new COBOL Table Selection dialog box will appear.

4. You will need to specify a valid table name for the COBOL record description.

5. Specify the A-Series COBOL source program file name, including **usercode** and **packname.**

6. Select the appropriate option button to indicate where Open/A will need to start processing the COBOL source at.

There are four choices for where Open/A may start processing the COBOL source. They are as follows: FD Name, FD Sequence Number, 01 Level Name, 01 Level Sequence Number. Based upon selection of any one of these choices, the corresponding text will change in the box below the option to allow you to specify the name of the FD, the sequence number of the FD, the 01 level name, or the 01 level sequence number.

7. Specify the name of the A-Series data file to be used, including **usercode** and **packname.**

8. **For flat file update, you will need to specify the name of a custom access library which will be used to read and write the file.** You will need to specify the name of this access library which is to be used to update the data file on the "COBOL Table Selection" dialog box, in the box labeled "Access Library". **If no flat file access library specification exist, the default program is used to access the file in read only mode.** This access library option would be used if you wanted to customize a library to read and update a site specific data file.

Open/A will then read the specified COBOL source program, starting at the location that has been specified. The table name which was specified in the "COBOL Table Selection" dialog box will appear in the list of Open/A tables in the Open/A window.

Thus, you may edit this table, declare this table updateable, etc., like any other table in Open/A.

For KIOII files, Open/A will automatically determine if any field in the given table is a key field and will use that field as an access into the table to retrieve data.

FLAT DESCRIPTION FILE AND DATA FILE LAYOUT

A "flat" description file will need to exist in order to provide information for the administration program. This description file will need to indicate the database name, dataset name(s), the name(s) of the files which contain the layout of the data, the actual data file name(s), and if you are updating a flat file, you will need to specify the code file name of the custom access library which will be used to read and write the file. This description file name will need to be specified in the description file box in the "Table Selection" screen in the administration program. The specified database name and dataset name(s) will be used in the same manner as a DMSII database name and a DMSII dataset name is used in the administration program. That is, the specified database name will appear in the list of databases and the specified dataset name will appear in the dataset list in the administration program. Multiple entries may appear in this file in order to access different flat files/KIOII files. Thus, this description file may contain multiple references to the same data layout, but refer to different data files, and/or it may contain references to different data file layouts. This allows you to maintain one flat description file which will reference different data file layouts; or multiple flat description files may exist which will contain various data file layouts. The format of this flat description file, created via CANDE and having a file type of "**DATA**", is described below.

The actual **data file layout** will need to be defined in a flat file, created by using CANDE and having a file type of "**DATA**". Consequently, the contents of this file will be similar to a record description layout. The valid field types are alpha, packed, unpacked, real, boolean, and field. Packed and unpacked fields may include a sign. An alpha field type will be the same as an alpha field in DMSII; a packed field type will be the same as a number field in DMSII; an unpacked field type will be number digits, but in character form; a real will be the same as a DMSII real; a boolean will be the same as a DMSII boolean; and a field will be the same as a DMSII field. **Redefines are allowed and group items will need to be redefined. KIOII sets will need to be declared; they will need to be defined in the same order as in the KEYEDIOII file and the same number of sets will need to exist. In addition, the set reference will need to appear at the end of the data file. Duplicates, or no duplicates, may be specified on the sets; the default is no duplicates**.

Note: No sets are allowed for flat files.
The result of a flat file query will contain a RECORD_NUM field; do not specify this field in the data file layout for the flat file.

Flat Description File Layout

A database name will need to be specified in the file. Following this database name, a line of text that includes the dataset name, the name of the file containing the layout of the data, and the actual data file name will need to be specified for each flat file/KIOII file which is to be accessed via Open/A. For a flat file, you will need to specify the custom access library code file name if you will be updating the file. Include the usercode and packname in the file names. Comments are indicated by the % sign; lines of text may wrap around onto another line.

SYNTAX:

```
DB=<database name>;
DS=<dataset name>,<name of file containing layout of data>,<actual data file name>,
< object file name of the custom access library, if applicable>;
```

Example of a flat description file:

```
%START LIST OF DATABASES
DB=KIOIIDB;

DS=KIOF1,(TEST)TEST/DESC/KIO/F1 ON PACK,(TEST)TEST/KIO/F1 ON PACK;
DS=KIOF2,(TEST)TEST/DESC/KIO/F2 ON PACK,(TEST)TEST/KIO/F2 ON PACK;
DS=MFGTRN,(TEST)DESC/KIO/MFGTRN ON PACK,(TEST)TEST/MFGTRN ON PACK;

DB=FLATDB;
DS=F1DS,(TEST)DESC/FLAT/F1,(TEST)TEST/FLAT/TESTDATA ON PACK;
```

Data File Layout

The data file layout is described below. A TYPE reference will need to be specified which will be either KEYEDIOII or FLATFILE. The data layout will need to follow this reference. As indicated above, the field types will need to be alpha, packed, unpacked, real, boolean, or field; a filler is allowed. In addition, redefines and group fields are allowed; however, group fields will need to be redefined as shown below. **The set reference(s), applicable to KIOII files only, may appear on multiple lines; however, all other references will need to be on a single line. KIOII sets will need to be defined in the same order as in the KEYEDIOII file and the same number of sets will need to exist.** Note that comments are indicated by the % sign.

Syntax:

```
TYPE=<KEYEDIOII, FLATFILE>
<data layout>
<set reference(s)>
```

where <data layout> may consist of the following lines:

```
<field name>                                    <field type>;
<field name1> REDEFINES <field name>            <field type>;
```

where <set reference> may consist of the following:

```
<set name> SET KEY= (<key name>, <key name>, .....)    <optional PRIME>
<optional DUPLICATES clause>;
```

OR <set name> SET KEY=<key name> <optional PRIME> <optional DUPLICATES clause>;

NOTE: field name may be <=29 characters
set name may be <= 17 characters

Example 1:

Example of a data file layout for a KIOII file:

COBOL RECORD LAYOUT(C74):

```
01  I-REC.
    03   I-SEX                                        PIC 9.
    03   I-AGE                                        PIC 99.
    03   I-PHONE                                      PIC 9(7).
    03   I-CODE.
         05  I-DIV                                    PIC 9(2) COMP.
         05  I-DEP                                    PIC 9(2) COMP.
         05  I-SEC                                    PIC 9(2) COMP.
    03   I-NAME.
         05  I-FIRST-NAME                             PIC X(20).
            05  I-LAST-NAME                           PIC X(20).
    03  FILLER                                        PIC X(7).
```

KIOII DATA FILE LAYOUT FOR THE ABOVE COBOL RECORD:

```
TYPE=KEYEDIOII
I-REC                                          ALPHA(60);
I-SEX             REDEFINES I-REC              UNPACKED(1);
I-AGE                                          UNPACKED(2);
I-PHONE                                        UNPACKED(7);
I-CODE                                         PACKED(6);
     I-DIV        REDEFINES I-CODE             PACKED(2);
     I-DEP                                     PACKED(2);
     I-SEC                                     PACKED(2);
I-NAME                                         ALPHA(40);
     I-FIRST-NAME  REDEFINES I-NAME           ALPHA(20);
     I-LAST-NAME                             ALPHA(20);
SET1 SET KEY=I-PHONE  PRIME NO DUPLICATES;
```

Example of a data file layout for a KIOII file:

COBOL RECORD LAYOUT(C74):

```
01  MFGTRN-REC1.
   03  MFGTRN-KEY.
      05  MFGTRN-PROPNO.
         07  MFGTRN-NO                   PIC 9(14) COMP.
      05  MFGTRN-DATE                    PIC 9(6) COMP.
      05  MFGTRN-SEQ                     PIC 9(2) COMP.
      05  MFGTRN-RECORD-TYPE.
           07  MFGTRN-TYPE               PIC 9(2) COMP.
         07  MFGTRN-TYPE-SEQ             PIC 9(2) COMP.
   03  MFGTRN-CODE                       PIC X(4).
   03  MFGTRN-OPERATOR           PIC X(2).
   03  MFGTRN-REFERENCE                  PIC X(10).
   03  MFGTRN-DESCRIPTION                PIC X(20).
   03  MFGTRN-DESCRIPTION2               PIC X(30).
   03  MFGTRN-TOTOL                      PIC S9(5)V99 COMP.
   03  MFGTRN-NO-OF-CHARGES              PIC 9(2) COMP.
    03  FILLER                           PIC X(6).
```

KIOII DATA FILE LAYOUT FOR THE ABOVE COBOL RECORD:

```
TYPE=KEYEDIOII
MFGTRN-REC1      ALPHA (90);
  MFGTRN-KEY  REDEFINES MFGTRN-REC1              ALPHA(13);
     MFGTRN-PROPNO  REDEFINES MFGTRN-KEY         PACKED (14);
       MFGTRN-NO  REDEFINES MFGTRN-PROPNO        PACKED(14);
     MFGTRN-DATE                                  PACKED(6);
     MFGTRN-SEQ                                   PACKED(2);
     MFGTRN-RECORD-TYPE                          PACKED(4);
        MFGTRN-TYPE  REDEFINES MFGTRN-RECORD-TYPE  PACKED(2);
        MFGTRN-TYPE-SEQ                           PACKED(2);
   MFGTRN-CODE                                   ALPHA(4);
   MFGTRN-OPERATOR                               ALPHA(2);
   MFGTRN-REFERENCE                              ALPHA(10);
   MFGTRN-DESCRIPTION                            ALPHA(20);
   MFGTRN-DESCRIPTION2                           ALPHA(30);
   MFGTRN-TOTAL                                   PACKED(S7,2);
   MFGTRN-NO-OF-CHARGES                          PACKED(2);
   FILLER                                        ALPHA(6);
SET1  SET  KEY=(MFGTRN-PROPNO,MFGTRN-DATE,MFGTRN-SEQ,
              MFGTRN-TYPE,MFGTRN-TYPE-SEQ)  PRIME  NO DUPLICATES;
```

Example of a data file layout for a flat file:

```
TYPE=FLATFILE
N                                     UNPACKED(4);
A                                     ALPHA(20);
%
%
%
```

% **Example of a group item and the group item redefined in a data layout:**

```
%                      G GROUP
%                       (
%                        G1           ALPHA(2)
%                        G2           ALPHA(2)
%                       )
%         GROUP G REDEFINED BELOW IN A  DATA LAYOUT
                       %
                       G                ALPHA(4);
                       G1  REDEFINES  G    ALPHA(2);
                       G2               ALPHA(2);
```

Chapter 10

Running Multiple Versions of Open/A

COMS Direct Window

In order to allow multiple versions of Open/A on the A-Series using COMS direct window interface, you will need to create a new and unique COMS program, window, and agenda (refer to chapter 6, “COMS Direct Window Program”, for detailed information, if necessary). The station will need to have its default window changed to refer to the new window.

TCP/IP

In order to allow multiple versions of Open/A on the A-Series using TCP/IP, the TCP/IP port needs to be changed. To change the port on the A-Series, you will need to add a line of text to the file “OPENA/CONFIG”, located on the same usercode and pack as the Open/A files. The syntax of the line of text is as follows:

```
TCPIPPORT=XXXX
```

where XXXX is the new port number. For example, to change to port 3000, you will need to add the line TCPIPPORT=3000 to the “OPENA/CONFIG” file. After the TCP/IP program is run, you may wish to use the “AX STATUS” command to verify the port that is being offered. The default port number is 2048.

In addition, you will need to make a corresponding change on the PC to enable Open/A to find the correct A-Series server. You will need to add a line in the TCPIP section of the OPENA.INI file, located in the Windows directory. The syntax of the line in the OPENA.INI file is as follows:

```
PORTNO=XXXX
```

where XXXX is the same port number as specified on the A-Series. In our above example, the line would be PORTNO=3000.

HLCN

In order to allow multiple versions of Open/A on the A-Series using HLCN, a unique HLCN id will need to be specified. You will need to add a line of text to the “OPENA/CONFIG” file, located on the same usercode and pack as the Open/A files. The syntax of this line of text is as follows:

```
HLCNID=XXXX
```

where XXXX is the new id(between 1 and 250 inclusive). For example, to allow an HLCN id of 100, the line of text to be added is HLCNID=100. After the HLCN program is run, you may wish to use the "AX STATUS" command to verify the id that is being used.

In addition, you will need to make a corresponding change on the PC to enable Open/A to find the correct A-Series server. You will need to add a line of text in the HLCN section of the OPENA.INI file, located in the Windows directory. The syntax of the line of text in the OPENA.INI file is as follows:

```
ID=XXXX
```

where XXXX is the same number as specified on the A-Series. In our above example, the line would be ID=100.

Chapter 11

Logging Feature

The Open/A logging feature will automatically be invoked when an application invokes Open/A. The codefile, “OBJECT/OPENA/ENVIRONMENT” controls the logging and allows for the usage of “AX” commands. The actual file which contains the logging information is “OPENA/LOG”, located under the same usercode and pack as the rest of the Open/A files. Initially, all information will be logged. A log file may be transferred; thereby, providing for archiving of the Open/A log files. **A log analyzer program is also available which will create a print file of the information collected in the log file**.

AX Commands

AX **LOG ON**
-turns logging on

AX **LOG OFF**
-turns logging off

AX **LOG TRANSFER**
-provides for archiving of the Open/A log files; the renamed file will be “OPENA/LOG/<date>/<SN>

AX **LOG STATUS**
-indicates if logging is available; or, if logging is currently on or off

WRU -displays Open/A version information

Note that these commands may be used in the “Environment Command” dialog box of the administration program; thereby alleviating the need to manually locate the mix number of the environment manager program. Refer to chapter 3, “Open/A Administration Program”, section entitled “Environment Command”.

Loganalyzer Program

The loganalyzer program, “OPENA/LOGANALYZER”, will allow you to specify a specific task number for analysis purposes; otherwise, all information in the log file will be spooled to a print file. In addition, you may specify a particular log file to use for analysis purpose.

Syntax:

```
RUN OPENA/LOGANALYZER;VALUE=<connection task #>;FILE LOGFILE=<log file name>
```

Examples:

```
RUN OPENA/LOGANALYZER;VALUE=3580;
FILE LOGFILE=OPENA/LOG/011095/6

RUN OPENA/LOGANALYZER;LOGFILE=OPENA/LOG/011095/6

RUN OPENA/LOGANALYZER
```

Chapter 12

Running the Administration Program in Batch Mode

Administration Program in Batch Mode

The Administration program may be run in batch mode to allow for batch generation of table libraries. Batch generation of table libraries should be done only if the DMALGOL compiler is not available due to site security reasons, or if the Open/A administrator wishes to make the table library generation part of the database reorganization process. The batch table library generation is accomplished by passing a command file to the Administration program when it is executed. The first seventy two characters of the command file are actually used. For example, to run the Administration program and have it process the command file "MYCMDS", you would enter the following:

```
RUN OPENA/ADMIN;FILE CMDS=MYCMDS
```

A blank line or line beginning with a "%" is treated as a comma. All other lines must be one of the following:

SYNTAX

This command turns on the "Syntax" mode. Each command will be processed for correct syntax, but the table libraries will not be generated. Normally, this will be the first line in the command file for a syntax run.

SET DESCFILE=<description file name>

where <description file name> is the name of your DMSII description file. This command sets the name of the description file to the given name. All subsequent commands will refer to this description file until another "SET DESCFILE" command.

SET DATABASE=<database name>

where <database name> is the name of the DMSII database for which you would like to generate table libraries. The database must reside in the description file that was referenced in the "SET DESCFILE" command. All subsequent commands will refer to this database until the next "SET DATABASE" command.

SET TABLELIB NODE=<node>
Sets the target node for all subsequent table library compiles to the location specified by <node>. The default <node> is the name OPENATL under the usercode where Open/A was installed. For example, if Open/A was installed under usercode "OPENA", then the default node is "(OPENA)OPENATL". You would only change this setting if you were generating table libraries for another machine and do not want to overwrite the libraries for the machine that you are currently running on.

SET TABLELIB PACKNAME=<packname>
Sets the target pack for all subsequent table library compiles to the pack specified by <packname>. The default <packname> is the pack on which Open/A was installed. For example, if Open/A was installed on PACK, then the default pack is "PACK". You would only change this if you were generating table libraries for another machine and do not want to overwrite the libraries for the machine that you are currently running on.

GENERATE <dataset>
This command generates the table library for <dataset>. The specified dataset must be in the database named in the "SET DATABASE" command and the database must be in the description file specified in the "SET DESCFILE" command. The generated object file is placed on disk in the location specified by the "TABLELIB NODE" AND "PACKNAME" commands.

EXIT
This command will cause the execution of the current command file to terminate.

LOG = <log spec>
where <log spec> specifies the destination for status output. <log spec> may be either NONE, DISPLAY, REMOTE, PRINTER, or <filename>. The default is no output.

If a syntax error occurs during the processing of the command file, the execution will be terminated. If the Administration program was run from a terminal, then the error message will be sent to that terminal; otherwise, it will be DISPLAY'ed. If the Administration program encounters an error, the TASKVALUE will be non-zero after the Administration program terminates; otherwise, it will be zero.

Suppose that you want to generate table libraries for the datasets “DS1”, “DS2”, and “DS3” that are in the logical database “LDB1” in the description file “(PROD)DESCRIPTION/COMPANYDB ON PRODPACK”. You could accomplish this by executing the following command file. The table libraries will be placed under a different node so that they may be moved to the machine “PRODA11”. If the table libraries were being generated to use on this machine, you would not set the tablelib node and packname.

```
%FILE TO GENERATE DATASETS FOR LDB1
SET DESCFILE = (PROD)DESCRIPTION/COMPANYDB ON
PRODPACK
SET DATABASE = LDB1
SET TABLELIB NODE = (OPENA)PRODA11/OBJECT
SET TABLELIB PACKNAME = PACK
GENERATE DS1
GENERATE DS2
GENERATE DS3
```

Chapter 13

Access Security

Overview

Open/A allows several levels of security to control access to the A-Series data. All access is controlled via client configuration files. These files reside on the host and are maintained by the Windows based Open/A Administration Program. When a client connects to the Open/A host based component, it specifies which client configuration file is to be used. This determines which tables and access rights that the client will receive.

Tables

A-Series DMSII datasets are referred to as Tables in the SQL/ODBC environment. The client configuration file contains a list of tables that the given client may access. This list is built and maintained in the "Table Select" function of the Open/A Administration Program. By default, a table may only be accessed in inquiry mode and may not be updated. However, the administrator may edit the table and mark the table as being updatable (refer to the Table Editing section, in chapter 3, "Open/A Administration Program"). All tables that have been marked as update capable are shown as red in the Open/A table list. Whether a table is marked as update capable or not determines how the corresponding dataset is opened on the A-Series. Tables which may not be updated are opened INQUIRY, whereas tables which may be updated are opened UPDATE. Even though a table has been marked as update capable, the program which is attempting to do the update must be in the list of allowed programs and must be update capable. See the following discussion for a description of program access control.

Programs

Before we can discuss Open/A program level security, we must examine the way that programs are named under Windows. In Windows, each program has a module name. This name is typically the same as the name of the program's EXE file. For example, Microsoft Access has the name MSACCESS, which is the same as the executable file, msaccess.exe. This name is embedded in the exe file and therefore does not change if the executable file is renamed. For example, if the Access executable is renamed to "x.exe" and run, its module name will still be MSACCESS. This fact forms the basis for the Open/A program security. A user may not rename an executable to a different name in order to circumvent the Open/A security.

Open/A program level security may be either enabled or disabled in a given client configuration file. If the security is disabled, which is the default, then all programs are allowed to inquire on all tables in the available list. If security is enabled, only programs which have been added to the Application Program list are allowed to access the tables. In addition, each program is marked as to whether it is allowed to update tables or if it is restricted to inquiry only. As extra security against unauthorized access, a program attempting to update a table must be in the application program list and marked as update capable even if the program level security is disabled.

NOTE: You may open an existing client configuration file to make changes or create a new one.

Refer to chapter 3, "Open/A Administration Program", section "Application Editing" for information relative to application program editing and the section "Global Client Configuration File Settings" for application security.

Chapter 14

Table Libraries and Database Libraries

Table Libraries

By default, a table library is used to access a dataset when a user references a table in a SQL query. This DMALGOL table library, which contains DMSII verbs, is automatically created and compiled the first time the dataset is referenced. For those sites which have retitled their DMALGOL compiler and DMINTERFACE file, Open/A provides the capability of specifying the names of the various levels of the DMALGOL compiler and DMINTERFACE file. For each level, an entry is required in the file "OPENA/CONFIG", located under the same usercode and on the same pack as the Open/A object files. Each entry will need to be as follows:

```
DMALGOLXX=<name>
DMINTERFACEXX=<name>
```

where XX is the level of DMSII and <*name*> is the full name of the compiler or DMINTERFACE. For example, in order to specify the compiler and DMINTERFACE for the 40 DMSII level, you will need to have the following entries in the "OPENA/CONFIG" file:

```
DMALGOL40=*SYSTEM/DMALGOL/402 ON DISK.

DMINTERFACE40=*SYSTEM/DMINTERFACE/402 ON DISK.
```

NOTE: Extra spaces are not allowed and the compiler name and DMINTERFACE name must be terminated with a ".".

Table libraries will automatically be regenerated if necessary: i.e.: after a DMSII reorganization.

Database Libraries

If database libraries is set, one database library per user will exist. With this feature, which is an option, all client updates for a database will be performed in the same transaction. This option is set on the configuration file level via the Open/A Administration program by selecting "Global" and placing an "X" in the box entitled "Use Database Libraries".

APPENDIX A

Open/A DMSII Considerations

Introduction

There are several areas where the A-Series DMSII does not map exactly to ANSI standard SQL. Some of these areas and their impact on Open/A are discussed below.

DMSII Structure Names

Since ANSI SQL does not allow the "-" character in item names and its usage is quite common in DMSII, some translation occurs. When Open/A returns a list of tables or columns to a Windows ODBC application, it translates all "-" characters to "_". When the ODBC application references these items with the "_", the "_" will be translated back to its native form before being used. If you are creating your own SQL statements, the fields may be referenced with their original names, including the "-"; but, all ODBC applications, such as Microsoft Access, will refer to the items with the "_" character in the name.

DMSII Groups

Since most ODBC applications will bring back all columns in a particular table, it is redundant to return a group and its constituent fields. For this reason, group fields are not listed as columns in a table. This means that an application must access the individual fields in the group. This is usually not a problem and makes sense for a SQL application. Open/A provides an option which allows you to specify whether or not you wish to have the group field(s) returned to the calling application. This option is discussed in chapter 3, "Open/A Administration Program", sections "Table Selection", and "Default Options".

DMSII Sets and SQL WHERE Conditions

The SQL WHERE condition may be any arbitrary selection expression involving any columns in the table. Open/A will analyze this logic and select a DMSII set, if any match, and perform a linear selection on the remaining logic. There are several issues involved here. The first issue is if more than one set matches, Open/A does not know which one to use. If Open/A were the DBA, it would select the set that it knew would return the fewest records, which would then have to be searched in a linear manner. At some future time, Open/A will be able to do some type of cost-based query optimization; but, at this point, it selects the set which has the highest percentage of key fields matched. In addition, Open/A will eventually use the WHERE condition on a subset in order to determine if the subset may be used for the query; however, for now, it must rule out any subsets, unless you have selected a subset table through the administration program to use for the query (refer to chapter 3, “Open/A Administration Program”, section “Table Selection”).

Abort Handling

Open/A will rollback a transaction if the PC goes away and has not explicitly committed the transaction. A transaction is immediately committed when “AUTOCOMMIT” is set; thus, there is no rollback situation in this case.

APPENDIX B

Dr. DeeBee Spy

Dr. DeeBee Spy has been licensed for your site to aid in the diagnosis of any ODBC problems that you might encounter during your trial and/or usage of Open/A. It allows all ODBC calls to and from a given ODBC datasource to be traced to a log file which may be examined to determine the sequence of ODBC events that occurred. You are not required to install this software unless requested to do so by the Open/A support staff. However, feel free to install and use Dr. DeeBee Spy if your in-house ODBC development staff will benefit from its use.

To run Dr. DeeBee Spy, follow the steps below:

1. Install the Dr. DeeBee Spy diskette, if it is not already installed on your PC.
2. Double click on the Dr. DeeBee Spy icon to run it.
3. Select the proper datasource.
4. Look for a file in the Windows directory; it will be called drdeebee.log.

APPENDIX C

Functions

Open/A supports the following functions:

STRING FUNCTIONS

The string functions listed here are 1-based; that is, the first character in the string is character 1.

Function	Description
CONCAT(string_exp1, string_exp2)	Returns a character string that is the result of concatenating *string_exp2* to *string_exp1*.
IFFUNC(Boolean exp1,exp2,exp3**)**	Returns *exp2* if Boolean *exp1* is true; returns *exp3* if Boolean exp1 is false. *Exp2* and *exp3* may be null.
INSERT(string_exp1,start,length,string_exp2)	Returns a characer string where *length* characters have been deleted from *string_exp1* beginning at *start* and where *string_exp2* has been inserted into *string_exp1*, beginning at *start*.
LCASE(string_exp)	Converts all upper case characters in *string_exp* to lower case.
LEFT(string_exp,count)	Returns the leftmost *coun*t of characters of *string_exp*.

LENGTH(string_exp)	Returns the number of characters in *string_exp*, excluding trailing blanks.

Function	Description
LOCATE(string_exp1,string_exp2 [,start])	Returns the starting position of the first occurrence of *string_exp1* within *string_exp2*. The search for the first occurrence of *string_exp1* begins with the first character position in *string_exp2* unless the optional argument, *start,* is specified. If *start* is specified, the search begins with the character position indicated by the value of start. If st*ring_exp1* is not found within *string_exp2*, the value 0 is returned.
LTRIM(string_exp)	Returns the caracters of *string_exp*, with leading blanks removed.
REPEAT(string_exp,count)	Returns a character string composed of *string_exp* repeated *coun*t times.
REPLACE(string_exp1,string_exp2,string_exp3)	Replaces all occurrences of *string_exp2* in *string_exp1* with *string_exp3*.

Function	Description
RIGHT(string_exp,count)	Returns the rightmost count of characters of *string_exp*.
RTRIM(string_exp)	Returns the characters of *string_exp* with trailing blanks removed.
SPACE(count)	Returns a character string consisting of *count* spaces.
STR(number)	Returns a string representation of the value of a numeric expression.
SUBSTRING(string_exp,start,length)	Returns a character string that is derived from *string_exp* beginning at the character position specifIed by *start* for *length* characters.
UCASE(string_exp)	Converts all lower case characters in *string_exp* to upper case.
VAL(string_exp)	Returns the numeric value of a string of characters.

The following function is only used in the “WHERE” condition.
ie: WHERE STRINGX LIKE “<pattern>“

Function	Description
LIKE “pattern”	Used to compare two string expressions in an expression. The following table shows the wildcard characters that you can use with the *Like* operator and the number of digits or strings they match.

Character(s) in *pattern*	**Matches in *expression***
\	escape
%	Zero or more characters
#	Any single digit(0-9)
-	Any single character
[*charlist*]	Any single character in *charlist*
[!*charlist*]	Any single character not in *charlist*

A group of one or more characters (*charlist*) enclosed in brackets ([]) can be used to match any single character in expression and can include almost any characters in the ANSI character set. In addition to a single list of characters enclosed in brackets, *charlist* can specify a range of characters by using a hyphen (-) to separate the upper and lower bounds of the range.

Function	**Description**
Examples of the wildcard usages:	
LAST-NAME LIKE “B%”	Returns all occurrences where last-name begins with a “B”.
EXP1 LIKE “\%”	Returns all occurrences where EXP1 is a percent.
CODEX LIKE “T[A-C]##”	Returns data that begins with the letter “T” followed by any letter “A-C” and two digits.
PRODUCT-CODE LIKE “[!ACF]%”	Retuns all occurrences where the first character of product-code is not A, C, or F and followed by any character.

NUMERIC FUNCTIONS

Function	Description
CEILING(numeric_exp)	Returns the smallest integer greater than or equal to *numeric_exp*.
EXP(float_exp)	Returns the exponential value of *float_exp*.
FLOOR(numeric_exp)	Returns largest integer less than or equal to *numeric_exp*.
MOD(integer_exp1,integer_exp2)	Returns the remainder (modulus) of *integer_exp1* divided by *integer_exp2*.
PI()	Returns the constant value of pi as a floating point value.
POWER(numeric_exp,integer_exp)	Returns the value of *numeric_exp* to the power of *integer_exp*.
RAND([integer_exp])	Returns a random floating point value using *integer_exp* as the optional seed value. Note: The optional seed value will have no affect on the random number.

Function	Description
ROUND(numeric_exp,integer_exp)	Returns *numeric_exp* rounded to *integer_exp* places right of the decimal point.
SIGN(numeric_exp)	Returns an indicator or the sign of *numeric_exp.* If *numeric_exp* is less than zero, -1 is returned. If *numeric_exp* equals zero, 0 is returned. If *numeric_exp* is greater than zero, 1 is returned.
SQRT(float_exp)	Returns the square root of *float_exp*.
TRUNCATE(numeric_exp,integer_exp)	Returns *numeric_exp* truncated to *integer_exp* places right of the decimal point.

TIME and DATE FUNCTIONS

Function	Description
CURDATE( )	Returns the current date as a date value.
CURTIME( )	Returns the current local time as a time value.
CURRENT TIMESTAMP	Same as NOW().
DATEDIFF(date1, date2)	Returns the number of days between *date1* and *date2 (date2-date1)*
DAYNAME(date_exp)	Returns a character string containing the name of the day (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday) for the day portion of *date_exp*.
DAYOFMONTH(date_exp)	Returns the day of the month in *date_exp* as an integer value in the range of 1-31.
DAYOFWEEK(date_exp)	Returns the day to the week in *date_exp* as an integer value in the range of 1-7, where 1 represents Sunday.
DAYOFYEAR(date_exp)	Returns the day of the year in *date_exp* as an integer value in the range of 1 - 366.

Function	Description
HOUR(time_exp)	Returns the hour in *time_exp* as an integer value in the range of 0 - 23.
MAKEDATE(year, month, day)	Returns a date from the numeric values of year, month, day.
MAKETIME(hour, minute, second)	Returns a time from the numeric values of hour, minute, and second.
MINUTE(time_exp)	Returns the minute in *time_exp* as an integer value in the range of 0 - 59.
MONTH(date_exp)	Returns the month in *date_exp* as an integer value in the range of 1 - 12.
MONTHNAME(date_exp)	Returns a character string containing the name of the month (January - December) for the month portion of *date_exp*.
NOW( )	Returns current date and time as a timestamp value.
QUARTER(date_exp)	Returns the quarter in *date_exp* as an integer value in the range of 1 - 4.

Function	Description
SECOND(time_exp)	Returns the second in *time_exp* as an integer value in the range of 0 - 59.
WEEK(date_exp)	Returns the week of the year in *date_exp* as an integer value in the range of 1 - 53.
YEAR(date_exp)	Returns the year in *date_exp* as an integer value. The range is data source-dependent.

Appendix D
SQL Grammar

Open/A supports the following SQL grammar.

SQL Statements

delete-statement-searched ::=
DELETE FROM table-name [**WHERE** search-condition]

insert-statement ::=
INSERT INTO *table-name* [(*column-identifier* [, *column-identifier*]...)]
VALUES (*insert-value* [, *insert-value*]...)

select-statement ::=
SELECT *select-list*
FROM *table-reference-list*
[**WHERE** *search-condition*]
[**GROUP BY** *column-name* [,*column-name*]...]
[**HAVING** *search-condition*]
[*order-by-clause*]

statement ::=
delete-statement-searched | *insert-statement* | *select-statement* | *update-statement-searched*

update-statement-searched ::=
UPDATE *table-name*
SET *column-identifier* = {*expression* | **NULL**} [, *column-identifier* = {*expression* | **NULL**}]...
[**WHERE** *search-condition*]

Elements Used in SQL Statements

approximate-numeric-literal ::= *mantissa* E *exponent*
mantissa ::= *exact-numeric-literal*
exponent ::= [+ | **-**] *unsigned-integer*

between-predicate ::= *expression* [NOT] BETWEEN *expression* AND *expression*

character-string-literal ::= ‘{*character*}...’

column-alias ::= *user-defined-name*

column-identifier ::= *user-defined-name*

column-name ::= [*table-name* | *correlation-name*.]*column-identifier*

comparison-operator ::= < | > | <= | >= | = | <>

comparison-predicate ::= *expression comparison-operator expression*

correlation-name ::= *user-defined-name*

date-literal ::= 'date-value'

date-separator ::=/

date-value ::= years-value date-separator months-value date-separator
days-value

days-value ::= digit digit

digit ::= **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

dynamic-parameter ::= **?**

exact-numeric-literal ::=
[+ | -] {*unsigned-integer*[.*unsigned-integer*] | *unsigned-integer*. | .*unsigned-integer*}

expression ::= *term* | *expression* {**+** | **-** } *term*
term ::= *factor* | *term* {***** | /} *factor*
factor ::= [+ | -] *primary*
primary ::= *column-name*
| *dynamic parameter*
| *literal*
| *ODBC-scalar-function-extension*
| *ODBC-date-time-extension*
| *set-function-reference*
| **(***expression***)**

hours-value ::=digit digit

in-predicate ::= *expression* [**NOT**] **IN** {(*value* [,*value*]...)}

value ::= *literal* | *dynamic-parameter*

insert-value ::=
literal
| *dynamic-parameter*
| **NULL**

letter ::= *lower-case-letter* | *upper-case-letter*

like-predicate ::= *expression* [**NOT**] **LIKE** *pattern-value*
pattern-value ::= *character-string-literal* | *dynamic-parameter*

literal ::= *character-string-literal* | *numeric-literal*
| *date-literal* | *time-literal* | *timestamp-literal*

lower-case-letter ::= **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** | **n** | **o** | **p** | **q** | **r** | **s** | **t |**
u | **v** | **w** | **x** | **y** | **z**

minutes-value ::=digit digit

month-value ::= digit digit

null-predicate ::= *column-name* **IS** [**NOT**] **NULL**

numeric-literal ::= *exact-numeric-literal* | *approximate-numeric-literal*

ODBC-date-literal ::=ODBC-ext-esc-initiator **d** *date-literal ODBC-ext-esc-terminator*

ODBC-date-time-extension ::=
ODBC-date-literal
| *ODBC-time-literal*
| *ODBC-timestamp-literal*

ODBC-ext-esc-initiator ::= **{**

ODBC-ext-esc-terminator ::= **}**

ODBC-outer-join-extension ::= *ODBC-ext-esc-initiator* **oj** *outer-join ODBC-ext-esc-terminator*

ODBC-scalar-function-extension ::= *ODBC-ext-esc-initiator* **fn** *scalar-function ODBC-ext-esc-terminator*

ODBC-time-literal ::=*ODBC-ext-esc-initiator* **t** *time-literal ODBC-ext-esc-terminator*

ODBC-timestamp-literal ::=*ODBC-ext-esc-initiator* **ts** *timestamp-literal ODBC-ext-esc-terminator*

order-by-clause ::= **ORDER BY** *sort-specification* [,*sort-specification*]...
sort-specification ::= {*unsigned-integer* | *column-name*} [**ASC** | **DESC**]

outer-join ::= {*table-name* [*correlation-name*] | *outer-join*} **LEFT OUTER JOIN**
table-name [*correlation-name*] **ON** *search-condition*

predicate ::= *between-predicate* | *comparison-predicate* | *in-predicate* | *like-predicate* | *null-predicate*

scalar-function ::= *function-name* (*argument-list*)
(see Appendix C for a list of function-names and argument-lists)

seconds-value ::= digit digit

search-condition ::= *boolean-term* [**OR** *search-condition*]
boolean-term ::= *boolean-factor* [**AND** *boolean-term*]
boolean-factor ::= [**NOT**] *boolean-primary*
boolean-primary ::= *predicate* | **(***search-condition***)**

seconds-fraction ::= unsigned-integer

select-list ::= ***** | *select-sublist* [,*select-sublist*]...
select-sublist ::= *expression* [[**AS**] *column-alias*] | {*table-name* | *correlation-name*}**.***

set-function-reference ::= **COUNT**(*) | {**AVG** | **COUNT** | **MAX** | **MIN** | **SUM**} **(***expression***)**

table-identifier ::= *user-defined-name*

table-name ::= *table-identifier* | *owner-name***.***table-identifier*

table-reference ::= *table-name* [*correlation-name*] | *ODBC-outer-join extension*

table-reference-list ::= *table-reference* [,*table-reference*]...

time-literal ::= ‘time-value’

*time-separator ::=***:**

timestamp-literal ::= ‘timestamp-value’

timestamp-separator ::= (The blank character)

timestamp-value ::= date-value timestamp-separator
time-value[**.***seconds-fraction*]

time-value ::= hours-value time-separator minutes-value time-separator
seconds-value

upper-case-letter ::= **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q**
| **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z**

unsigned-integer ::= {*digit*...}

user-defined-name ::= *letter* [*digit* | *letter* | **_**]...

years-value ::= digit digit digit digit

APPENDIX E

DATE and TIME FORMATS

Open/A supports the following date formats:
NOTE: These time/timestamp formats are relative to the
time/timestamp formats on the A-SERIES.
Refer to the A-SERIES ALGOL reference manual.

DATE FORMATS

CENTURY
YYDDD
YYYYDDD

MMDDYY
MMDDYYYY
MMYYDD
MMYYYYDD
MM/DD/YY
MM/DD/YYYY
MM/YY/DD
MM/YYYY/DD

DDMMYY
DDMMYYYY
DDYYMM
DDYYYYMM
DD/MM/YY
DD/MM/YYYY
DD/YY/MM
DD/YYYY/MM

YYMMDD
YYDDMM
YY/MM/DD
YY/DD/MM
YYYYMMDD
YYYYDDMM
YYYY/MM/DD
YYYY/DD/MM

*TIME(6)
*TIME(7)
*TIME(10)
*TIME(15)
*TIME(16)

*These time formats contain HH:MM:SS.

TIME FORMATS

Open/A supports the following time formats:

Time Formats:

HHMMSS
TIME(1)
TIME(6)
TIME(7)
TIME(9)
TIME(11)
TIME(16

Timestamp Formats:

(contain both a date and a time)

TIME(6)
TIME(7)
TIME(16)
HHMMSSFF

APPENDIX F

DMSII TO ODBC DATATYPE MAPPINGS

DMSII	ODBC
--------	---------
GROUP	SQL_VARCHAR
GROUP(FIELD OR BOOLEAN)	SQL_INTEGER
BOOLEAN	SQL_BIT
FIELD	SQL_INTEGER
ALPHA	SQL_VARCHAR
NUMBER(SI)	
I > 9	SQL_DOUBLE
I <= 4	
SQL_SMALLINT	
I>4<=9	SQL_INTEGER
NUMBER(I)	
I > 9	SQL_DOUBLE
I <=4	
SQL_SMALLINT	
I>4<=9	SQL_INTEGER
NUMBER(SI,J)	
I<= 9	SQL_NUMERIC
I > 9	SQL_DOUBLE
NUMBER(I,J)	
I <= 9	SQL_NUMERIC
I > 9	SQL_DOUBLE

REAL	SQL_DOUBLE