The OLR System
Application Programming Interface (The OLR API)

Release 3.0

Programmer's Guide

Contents

Preface About This Guide

This guide is for application designers and programmers who will use the services of the OLR System API to add document services or concurrent processing services to applications.

Chapter 1 provides an overview of the OLR API and the services it provides.

Chapter 2 discusses the OLR Attach Facility. Read this chapter for the information you'll need to design and develop help windows, notepads and reference libraries that can be controlled using the business rules of your application.

Chapter 3 discusses the OLR Intercept Attach Facility. Read this chapter for the information you'll need to design and develop applets that can be invoked using function keys that you define to the OLR System Intercept Facility.

Chapter 4 describes the OLR Server, its structure and contents. Use this information to design and program customized data access and update routines using the OLR Server.

Chapter 1 OLR API Features

This chapter describes the features of the OLR Application Programming Interface (OLRAPI):

- the **OLR Attach Facility**, which enables you to attach OLR System services to your applications
- the **Intercept Attach Facility**, which enables you to attach applets that you develop to your applications using function keys you define using the OLR System intercept service
- the **OLR Server**, which manages group libraries used by OnLine Help, OnLine Reference and OnLine Notepad. Licensed users of the OLR API can develop SQL to access and update information in the OLR Server and have access to a library of available data access routines.

The OLR System Attach Facility

The OLR System Attach Facility allows you to invoke the services of OLR System from within your CICS application.

The standard interface provided with the OLR System product enables direct access to the OnLine Reference product from a CICS command prompt, and hot key access to OnLine Help and OnLine Reference from a CICS transaction.

The standard interface includes a number of customization exit points that you can use for security checking, to change the topic or group that will display, to create an activity log, or to dynamically create a topic at user request time. These services can be provided independent of the business rules in your application, and do not require the OLR API.

Uses of the OLR System Attach Facility

If your business requirements call for your application to have a greater level of control over the behavior of the OLR System than is provided by the standard interface, the OLR Attach Facility should be considered.

Some examples of OLR Attach Facility uses are:

- displaying a help window automatically if a certain value is entered in a field, a new user signs onto an application, or urgent information needs to be displayed.
- displaying a note list over the lower half of a screen if a business record is requested and that record has notes attached of a specific priority, title, or date.
- invoking the OLR System at a specific topic or in a specific group from a menu transaction.
- requesting and potentially requiring a user to enter a note when a certain condition occurs, a decision is made, or a transaction is entered.

Implementing the OLR System Attach Facility

You can implement the Attach Facility using either the CICS Start verb or the CICS XCTL. Once the OLR System service has been given control, the user has the ability to display as many screens as needed and use any authorized functions before returning.

When the user presses the return key (typically F3) to terminate OLR System processing, the interface control block is returned in the COMMAREA. Your program is responsible for re-displaying the application screen at that point, and can use the information available in the COMMAREA to determine how to proceed.

For more information on the use of the OLR System Attach Facility, see chapter 2.

The Intercept Attach Facility

The Intercept Attach Facility enables you to use the services of the OLR System Intercept to attach programs you write as hot key options to your applications.

The standard interface to the OnLine Help and OnLine Notepad features of the OLR System uses a generalized transaction intercept facility to:

- evaluate function keys pressed on incoming messages,
- determine whether the hot key is requesting a service external to the application program
- suspend the processing of the application program when an external function is requested
- activate the external function and watch for it to complete, then
- resume the transaction, preserving all data entry values and attributes that have changed.

The Intercept Attach Facility enables you to develop and implement custom user services based on hotkeys that you define to the OLR System Intercept facility.

Uses of the Intercept Attach Facility

If your business requirements call for the user to have information on the screen from more than one file or business record at a time, or to use pop-up services such as calculators or record locators, you should consider using the Attach Facility, particularly if these services should be made available to a number of transaction screens.

Some examples of OLR Intercept Attach Facility uses are:

- displaying additional customer information in a pop-up window while using a transaction which displays only a limited amount of this information
- providing an applet which calculates a price quote based on information gathered from the active screen and information provided to the customer service representative by the customer.
- bringing information from an external technical environment to the desktop of a CICS user.

Implementing the Intercept Attach Facility

You enable a program to be initiated by the Intercept Attach Facility by:

- defining the transaction which will use the hot key function to the OLR System intercept (OLRM transaction) and assigning the USER function to a function key
- writing or modifying your Intercept Attach Facility User Exit Program to provide information about the program you want to invoke
- writing the applet you want to invoke, or tailoring the existing program you want to enable with a hot key, then defining it to the CICS region.

For more information on the use of the Intercept Attach Facility, see chapter 3.

The OLR Server

The OLR Server is a DB2 database which manages user access rules, group libraries, topics, notes and keywords used by the OLR System.

OnLine Help, OnLine Reference and OnLine Notepad access and update the OLR Server using SQL. In some cases, service routines are used to compress or decompress data, expand outlines, or transfer data from one copy of the server to another.

Licensed users of the OLR API can develop custom programs to access and update the contents of OLR Server independent of the functions provided by OnLine Help, OnLine Reference OnLine Notepad, and the OLR System Import/Export facility.

We advise customers who have business requirements that can be met using the OLR Server to contact our Technical Services group for advice on the best approach for organizing the data, implementing access controls, and writing programs to use the database.

Uses of the OLR Server

Some examples of the use of OLR Server as a user database are:

- generating content directly to the server from business activities (for example, generating an automatic note when unusual activity is detected against an account)
- translating the content of the server to an external format programmatically
- storing object data on a copy of the OLR Server tables for use in a custom application.

Implementing Applications using the OLR Server

We advise customers to work with our Technical Services staff during the design and implementation of applications which use the OLR Server. As a licensed user of the OLR API, you have the ability to request and receive the information you need to design applications and to obtain additional technical information as needed.

.

Chapter 2 The OLR Attach Facility

The Attach Facility of the OLRAPI allows you to invoke the OLR System from your CICS applications. You may use this facility to:

- include the OLR System as a menu choice from one of your own application screens, returning to the menu program when the OLR session is ended.
- bring up a help window based on the criteria passed from your application.
- bring up the Notepad Options window based on the criteria passed from your application.
- bring up a half-screen note window allowing the user limited access to add, read or update a specific note under the control of your application program.

The OLR Interface Block

Communication between the OLR System and your application programs takes place through its application programming interface called the OLR Interface Block (OIB). Parameters passed through the OIB tell the OLR System what kind of attach is being requested and provide key information needed to process the request.

The OLR System uses the OIB to return status information about the processing of the request. The OIB is described in detail on the following page. The **OIB001** member in the **COPIES** dataset from the install tape contains COBOL copy code for the OIB. The **OIB001B** member in the **COPIES** dataset contains ASSEMBLER copy code for the OIB.

Attaching OLR from your CICS application

The OLR System may be attached from a user CICS program via START or XCTL.

Use the following CICS program control statement to attach the OLR System via START:

```
EXEC CICS START
     TRANSID  ('OLRX')
     FROM     (OIB-RECORD)
     LENGTH   (oib-record-length)
     TERMID   (EIBTRMID)
END-EXEC.
```

Use the following CICS program control statement to attach the OLR System via XCTL:

```
EXEC CICS XCTL
     PROGRAM  ('DBXPR000')
     COMMAREA (OIB-RECORD)
     LENGTH   (oib-record-length)
END-EXEC.
```

Control will be returned to your application program using the method indicated in the OIB (see OIB-RTN-MODE). If the return is via start, the OIB will be returned as a data record, to be retrieved by your program using a CICS RETRIEVE. If return is via XCTL, the OIB will be returned in the COMMAREA. Your application is responsible for the restore and redisplay of your application screen when you regain control.

The OLR Interface Block (OIB)

Field	Contents
Record Length	The total length of the OIB-RECORD, including the OIB-USER-AREA (see below). Length of the OIB-RECORD without a user area is 292 bytes.
Record ID	Must be 'DBXOIB'
Return Code	Returned by OLR to indicate the success of the attach request. Possible values are: **Normal Return** - no errors. **Key Failure** - If request was for NOTE-AUTO-ADD, this return code indicates that the note could not be added because it already existed. If request was for NOTE-READ or NOTE-UPDATE, the request could not be completed because the note could not be found. If request was for NOTE-LIST, note list is not displayed because either the topic could not be found or there are no notes attached to the topic in a group. **Failure** - a fatal error was encountered. See OIB-MESSAGE for information about the error that occurred.
Message	If the return code is not Normal Return, this field will contain a message describing the error encountered.
Request Type	Passed by user program to indicate type of attach requested. Required. Possible values are: **Admin Menu, Reference, Help List** - invoke the OLR System at the initial screen. **Reference** - display the OnLine Reference Select Book screen **Field Help** - display a pop-up help window based on help criteria provided. User will have access to same functions available when invoking help via the CICS Intercept. **Help List** - display the OnLine Help Select Help Link screen **Notepad** - display the OnLine Notepad Options window based on notepad criteria provided. User will have access to same functions available when invoking notepad via the CICS Intercept. **Note Auto-add** - Add a Note using the criteria provided, then display a half-screen edit window for that note. **Note List** - display the note list window based on the criteria provided. **Note Read** - display the text of a note in a half screen window using the criteria provided. **Note Update** - display the text of a note in a half screen window using the criteria provided and enable the user to update it.

Field	Contents
Return Mode	Passed by user program to indicate how control is to be returned from OLR. Required. Possible values are: **Return via XCTL** - control is to be returned via XCTL. OIB will be passed in COMMAREA. **Return via START** - control is to be returned via START. OIB will be returned as a data record, to be retrieved by the user program using a CICS RETRIEVE.
Return Program	Passed by user program. Required: If Return Mode is XCTL, this field should contain the name of the program to which control is to be transferred. If Return Mode is START, this field should contain the 4 character ID of the transaction to be started.
Extended Data Stream Flag	Passed by user program. Set to indicate whether or not the screen of the calling application uses an extended datastream. (Color, etc.)
Alternate Screen Size Flag	Passed by user program. Set to indicate whether or not the calling application uses an alternate screen size.
Bypass Key-Failure Message Indicator	Passed by user program. Set to indicate whether or not OLR will display a key-failure message window for the Note Auto-Add, Note Read, Note Update and Note List requests.
Restore Screen Indicator	This is currently not used.
Topic Group ID	Passed by user program. Used only with Notepad and Note Auto-add requests. If the topic a note is to be added to as the result of one of the above requests does not already exist on the database, the topic will be added automatically to the Group provided here.
Note Group ID	Passed by user program. Used only with Note Auto-add and Note List requests. For a Note Auto-Add request, when a note is added as the result of the above request, it is added automatically to the Group provided here. For a Note List request, this is optional. If a Group is provided, this is used to check for notes attached to a topic. Otherwise, the user's registered note group will be used.
Application ID	Passed by user program. Used only with Field Help requests. Provides the Application ID of the Help Link to be retrieved.
Screen ID	Passed by user program. Used only with Field Help requests. Provides the Screen ID of the Help Link to be retrieved.
Cursor Position	Passed by user program. Used only with Field Help requests. Used to determine the row and column of the Help Link to be retrieved.
Userid	Passed by user program. If provided, will be used to identify the user for the OLR session. Otherwise, the user-id will be derived by the usual means. (as set in the OLR System Parameters)

Field	Contents
Topic Title	Passed by user program. Used only with Notepad, Note Auto-add and Note List requests. For the Notepad and Note Auto-add requests, if the topic a note is to be added to does not already exist on the database, the topic will be added automatically using the title provided here. For the Note List request, if the topic provided whose list of notes will be displayed, a key-failure error will occur if it does not already exit on the database.
Topic Qualifier	Passed by user program. Used only with Notepad, Note Auto-add and Note List requests. For the Notepad and Note Auto-add requests, if the topic a note is to be added to does not already exist on the database, the topic will be added automatically using the title provided here. For the Note List request, if the topic provided whose list of notes will be displayed, a key-failure error will occur if it does not already exit on the database.
Note Title	Passed by user program. Used only with Note Auto-add, Note Read, and Note Update requests. When a note is added as the result of the above request, is added automatically using the title provided here.
Note Qualifier	Passed by user program. Used only with Note Auto-add, Note Read, and Note Update requests. When a note is added as the result of the above request, is added automatically using the qualifier provided here.
Flash Form Title	Passed by user program. Used only with Note Auto-add. When supplied, provides the title of the form that will be displayed.
Flash Form Qualifier	Passed by user program. Used only with Note Auto-add. When supplied, provides the qualifier of the form that will be displayed.
Form Field Value QID	When a form is requested, this field provides the optional 8 byte identifier of a CICS TSQ which contains initial values for the form fields to be displayed.
Form Indicator	When set to 'Y', OnLine Notepad will process the form specified above. If not 'Y', OnLine Notepad will display a blank screen for the Note.
User Area	A user area may be concatenated to the bottom of the OIB to allow you to hold information across the attach to OLR. If a user area is included on the OIB record when OLR is invoked, it will be returned to the user program unchanged when control is returned.
User Area Length	Passed by user program. This field should be set to reflect the length of the user area (including the User Area Length field) If a user area is used, the OIB-REC-LENGTH should be set to reflect the length of the entire OIB-RECORD, including the user area. The length of the OIB-RECORD (including the user area) may not exceed 4144 bytes.

Chapter 3 The Intercept Attach Facility

The Intercept Attach Facility of the OLR API allows you to use the OLR CICS Intercept to invoke applications you've written when the user presses a function key you specify.

Using this feature, a user working with one online application can hot key into another transaction (in the same way that a user can go into OnLine Help or OnLine Notepad at the touch of a PF key), returning to the original application when done.

Attaching applications via the OLR CICS Intercept

To attach a different application from an active application using the OLR CICS Intercept:

- Define the application to be intercepted to the OLR CICS Intercept using the OLRM transaction, in the same way that you would define an application if you wanted to connect it to OnLine Help or OnLine Notepad. (See Chapter 2 of the Administrator's Guide for information about this process).

  When defining first level criteria for the application to be intercepted, type **USER** for the function, rather than **HELP** or **NOTE**.

- Write an IAF User Exit Program (DBXUXIAF). When a **USER** type intercept occurs, this exit program is invoked to provide information about the target application. The target application is the application that the intercept passes control to. See below for more information about writing an IAF User Exit Program.

- Add a program definition for DBXUXIAF on the CICS region where the OLR System is installed.

- Tailor the target application to handle being invoked by the OLR Intercept, and to return control to the intercept when done. See below for more information about target application considerations.

Writing an IAF User Exit Program

The IAF User Exit Program (DBXUXIAF) is a command-level CICS program written at your site. It is accessed via CICS link whenever a **USER** type intercept has occurred. A skeleton version of DBXUXIAF is included in the **SAMPLES** dataset on the installation tape.

Input parameters giving extensive information about the context of the call are passed to the program through the COMMAREA. The IAF User Exit Program uses the COMMAREA to return information telling the Intercept how to attach the target application.

The member **IAF001** in the **COPIES** dataset on the installation tape contains a commarea copy definition for an IAF User Exit Program written in COBOL. Use **IAF001B** for an IAF User Exit Program written in Assembler.

DBXUXIAF parameters (IAF001)

Field	Contents
Parm Length	The total length of IAF001 parameter record. (374)
Exit Name	Must be 'DBXUXIAF'
Return Code	Returned by DBXUXIAF to indicate the success of the exit call. Possible values are: **Normal Return** - no errors. **No Intercept** - the transaction should not be intercepted. When this return code is used, the user will see a message and the original application session will be re-established when the user presses a function key to clear the message. **Fail With Dump** - an unexpected condition was encountered. A dump will be generated (using RETURN REASON as the dump code). The user will see a message and the original application session will be re-established when the user presses a function key to clear the message.
Return Reason	If the return from DBXUXIAF indicates FAIL-W-DUMP, the value in this field will be used as a dump code.
Type Attach	Returned by DBXUXIAF to indicate how the target application is to be attached. Possible values are: **XCTL** - the target application is to be invoked via XCTL. **START** - the target application is to be invoked via START. A START is required in MRO situations where the target application is not in the same region as the OLR System. While it is up to DBXUXIAF to indicate how the target application is to be invoked, the Intercept will handle the actual XCTL or START.
Backdrop Indicator	Returned by DBXUXIAF to indicate whether the OLR Intercept should display a protected version of the intercepted application screen before passing control to the target application. Possible values are: **Send Backdrop** - a protected version of the intercepted application screen will be displayed. **Don't Send** - the intercepted application screen will be left as is.
Attach Program	Returned by DBXUXIAF. Required: If TYPE-ATTACH is XCTL, this field should contain the name of the program to which control is to be transferred. If TYPE ATTACH is START, this field should contain the 4 character ID of the transaction to be started.

Field	Contents
Intercept Information	Passed to DBXUXIAF by OLR Intercept. These fields contain information about the intercepted transaction: **Transaction ID** **Cursor Row/Column** - position of cursor when PF Key was pressed **Field Row/Column/Length** - information about the field at the cursor when PF Key was pressed. **Intercept Queue ID** - ID of the queue where the image of the intercepted screen is stored. Used when calling the DBXEXTDS utility to extract information about the content of fields on the intercepted screen. (see Administrator's Guide for information about using DBXEXTDS). **PF Key invoking intercept** The information in these fields will also be passed to the target application.
Intercept Information (Modifiable)	Passed to DBXUXIAF by OLR Intercept. These fields contain information about the intercepted transaction: **Userid** **Application ID** **Screen ID** **Group ID** **Field Value** - content of the field the cursor was on when the intercept PF Key was pressed, up to 50 characters. **Field Qualifier** - not currently used. Always spaces. The information in these fields will also be passed to the target application. The values provided may be overridden by DBXUXIAF.
User Area	Returned by DBXUXIAF. This 128 byte block will be passed to the target application

Target Application Considerations

The target application is attached either by XCTL or START, depending on the TYPE-ATTACH requested by the IAF User Exit Program. The OLR Intercept passes a parameter list to the target application. In the case of an XCTL, the parameter list is passed in the COMMAREA. When attached via a START, the parameter list is passed as a data record, to be retrieved by the target application using a CICS RETRIEVE.

The member **IAF002** in the **COPIES** dataset on the installation tape contains a copy definition for the intercept parameters for a target application written in COBOL. Use **IAF002B** for a target application written in Assembler. These parameters are strictly for the use of the target application. They contain information about the context in which interception occurred, including a USER AREA passed from the IAF User Exit Program. They do not need to be saved or returned to the OLR System when the original application session is restored.

The target application returns control to the OLR Intercept when it is time for the original application session to be restored. This is done by issuing a START on the RETURN TRANID passed to the target application in the intercept parameters. This can be accomplished using the following CICS command:

```
EXEC   CICS START
       TRANSID   (return-tranid)
       TERMID    (EIBTRMID)
END-EXEC.
```

Intercept parameters for target application (IAF002)

Field	Description
Parm Length	The total length of IAF002 parameter record. (314)
Record Name	Must be 'DBA IAF'
Intercept Information	Passed to target application by OLR Intercept. These fields contain information about the intercepted transaction: **Transaction ID** **Cursor Row/Column** - position of cursor when intercept PF Key was pressed **Field Row/Column/Length** - information about the field the cursor was on when intercept PF Key was pressed. **Intercept Queue ID** - ID of the queue where the image of the intercepted screen is stored. Used when calling the DBXEXTDS utility to extract information about the content of fields on the intercepted screen. (see Administrator's Guide for information about using DBXEXTDS). **PF Key invoking intercept**
Return Tranid	Passed to target application by OLR Intercept. This field contains the ID of the transaction to be used by the target application to restore the original application session.
More Intercept Information	Passed to target application by OLR Intercept. These fields contain information about the intercepted transaction: **Userid** **Application ID** **Screen ID** **Group ID** **Field Value** - content of the field the cursor was on when the intercept PF Key was pressed, up to 50 characters. **Field Qualifier** - not currently used. Always spaces. The information in these fields may have been modified by DBXUXIAF.
Intercept Status	Passed to target application by OLR Intercept. These fields contain information about the status of the intercept and the terminal: **Intercept type** - GENERIC or BMS **Terminal Case** - Upper or Lower **Extended Datastream Indicator** **Alternate Screen Size Indicator** **Screen columns/lines** **OLR indicator** - indicates whether the intercepted transaction was part of the OLR System **ATI indicator** - indicates whether a START can be performed on the intercepted terminal.
IAF Parms	User Area passed from DBXUXIAF.

Chapter 4 The OLR Server

The OLR Server manages group libraries containing topics and notes, hypertext links between them and into them from application screens, and user authorization information which is used to control access and update rights.

Licensed users of the OLR API can write programs to load data into the OLR Server, convert the content from one format to another, or extract information.

This chapter describes the entities and relationships managed by OLR Server. Additional information is available from the Technical Services group as needed.

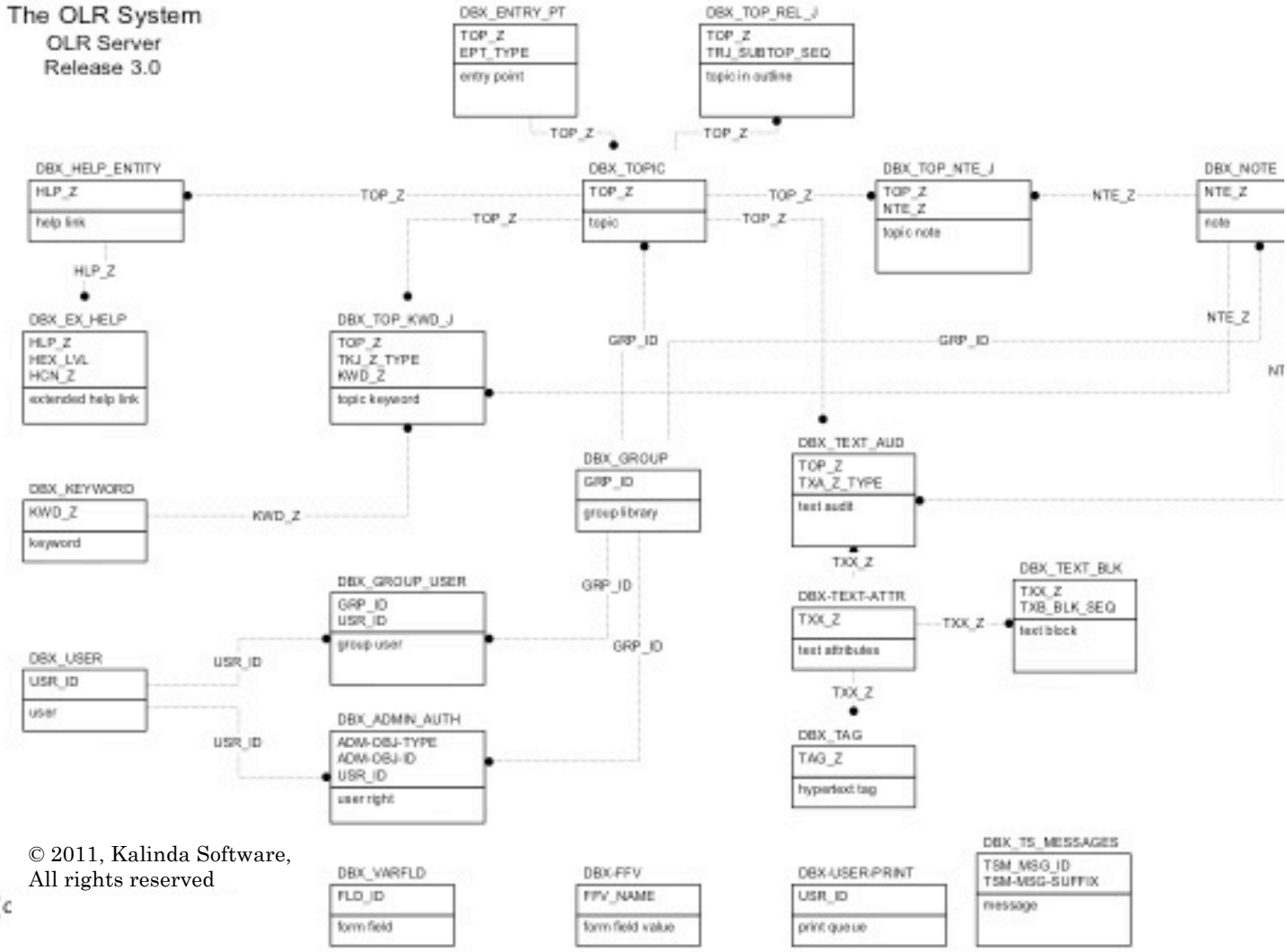
If you are planning to develop an application using the OLR Server, we advise you to contact us and discuss with us the best approach.

OLR Server Diagram

The diagram on the following pages depicts the entities and relationships managed by the OLR Server.

Table declarations follow the diagram.

DBA Software has a library of pre-build data access routines to search for topics or notes, explode outlines, check user authorizations and perform other functions used by the OLR System. If your application requires any of these routines, contact Technical Services for a current copy.



OLR Server Table Declarations

Table	Declaration
DBX_ADMIN_AUTH	EXEC SQL DECLARE DBX_ADMIN_AUTH TABLE (ADM_OBJ_TYPE CHAR(3) NOT NULL, ADM_OBJ_ID CHAR(10) NOT NULL, USR_ID CHAR(8) NOT NULL, ADM_GRANT_AUTH CHAR(1) NOT NULL, ADM_ADD_USER CHAR(8) NOT NULL, ADM_ADD_TMDT TIMESTAMP NOT NULL, ADM_UPDT_USER CHAR(8) NOT NULL, ADM_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_ENTRY_PT	EXEC SQL DECLARE DBX_ENTRY_PT TABLE (TOP_Z TIMESTAMP NOT NULL, EPT_TYPE CHAR(1) NOT NULL, EPT_ADD_USER CHAR(8) NOT NULL, EPT_ADD_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_EX_HELP	EXEC SQL DECLARE DBX_EX_HELP TABLE (HLP_Z TIMESTAMP NOT NULL, HEX_LVL SMALLINT NOT NULL, HCN_Z TIMESTAMP, TOP_Z TIMESTAMP NOT NULL, HEX_TYPE CHAR(1) NOT NULL, HEX_WNDW_ROW SMALLINT NOT NULL, HEX_WNDW_COL SMALLINT NOT NULL, HEX_ADD_USER CHAR(8) NOT NULL, HEX_ADD_TMDT TIMESTAMP NOT NULL, HEX_UPDT_USER CHAR(8) NOT NULL, HEX_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_FFV	EXEC SQL DECLARE DBX_FFV TABLE (FFV_NAME CHAR(32) NOT NULL, FFV_ROW SMALLINT NOT NULL, FFV_COL SMALLINT NOT NULL, FFV_OCC_ABS SMALLINT NOT NULL, FFV_OCC_NAME SMALLINT NOT NULL, FFV_LEN SMALLINT NOT NULL, FFV_TAGID CHAR(2) NOT NULL, FFV_DISP_TYPE CHAR(1) NOT NULL, FFV_REQ CHAR(1) NOT NULL, FFV_EDITRULE CHAR(8) NOT NULL, FFV_VALUE CHAR(50) NOT NULL, FFV_ADD_USER CHAR(8) NOT NULL, FFV_ADD_TMDT TIMESTAMP NOT NULL, FFV_UPDT_USER CHAR(8) NOT NULL, FFV_UPDT_TMDT TIMESTAMP NOT NULL, FFV_TXX_Z TIMESTAMP NOT NULL, FFV_TOP_Z TIMESTAMP NOT NULL,

Table	Declaration
	FFV_NTE_Z TIMESTAMP NOT NULL, FFV_FORM_Z TIMESTAMP NOT NULL) END-EXEC.
DBX_GROUP	EXEC SQL DECLARE DBX_GROUP TABLE (GRP_ID CHAR(10) NOT NULL, GRP_READ_PSSWD CHAR(8) NOT NULL, GRP_UPDT_PSSWD CHAR(8) NOT NULL, GRP_HELP_PSSWD CHAR(8) NOT NULL, GRP_ADMIN_PSSWD CHAR(8) NOT NULL, GRP_RSTRCT_IND CHAR(1) NOT NULL, GRP_DSPLY_IND CHAR(1) NOT NULL, GRP_ADD_USER CHAR(8) NOT NULL, GRP_ADD_TMDT TIMESTAMP NOT NULL, GRP_UPDT_USER CHAR(8) NOT NULL, GRP_UPDT_TMDT TIMESTAMP NOT NULL, GRP_READ_PROT_IND CHAR(1) NOT NULL, GRP_HRC_RSTRCT_IND CHAR(1) NOT NULL, GRP_ALS_RSTRCT_IND CHAR(1) NOT NULL, GRP_CNC_RSTRCT_IND CHAR(1) NOT NULL, GRP_DESC CHAR(25) NOT NULL) END-EXEC.
DBX_GROUP_USER	EXEC SQL DECLARE DBX_GROUP_USER TABLE (GRP_ID CHAR(10) NOT NULL, USR_ID CHAR(8) NOT NULL, GRP_ID_NOTE CHAR(10), GRU_TOP_LIST_AUTH CHAR(1) NOT NULL, GRU_TOP_UPDT_AUTH CHAR(1) NOT NULL, GRU_HRC_UPDT_AUTH CHAR(1) NOT NULL, GRU_NTE_LIST_AUTH CHAR(1) NOT NULL, GRU_NTE_UPDT_AUTH CHAR(1) NOT NULL, GRU_HLP_UPDT_AUTH CHAR(1) NOT NULL, GRU_READ_PROT_AUTH CHAR(1) NOT NULL, GRU_KWD_INDX_AUTH CHAR(1) NOT NULL, GRU_LAST_ACC_TMDT TIMESTAMP, GRU_ADD_USER CHAR(8) NOT NULL, GRU_ADD_TMDT TIMESTAMP NOT NULL, GRU_UPDT_USER CHAR(8) NOT NULL, GRU_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_HELP_ENTITY	EXEC SQL DECLARE DBX_HELP_ENTITY TABLE (HLP_Z TIMESTAMP NOT NULL, HLP_APPL_ID CHAR(8) NOT NULL, HLP_SCRN_ID CHAR(8) NOT NULL, HLP_ROW SMALLINT NOT NULL, HLP_COLUMN SMALLINT NOT NULL, HLP_LENGTH SMALLINT NOT NULL, TOP_Z TIMESTAMP NOT NULL, HLP_ADD_USER CHAR(8) NOT NULL, HLP_ADD_TMDT TIMESTAMP NOT NULL, HLP_UPDT_USER CHAR(8) NOT NULL, HLP_UPDT_TMDT TIMESTAMP NOT NULL, HLP_Z_TYPE CHAR(3) NOT NULL,

Table	Declaration
	HLP_TYPE CHAR(1) NOT NULL, HLP_WNDW_ROW SMALLINT NOT NULL, HLP_WNDW_COL SMALLINT NOT NULL) END-EXEC.
DBX_KEYWORD	EXEC SQL DECLARE DBX_KEYWORD TABLE (KWD_Z TIMESTAMP NOT NULL, KWD_STRING CHAR(32) NOT NULL, KWD_DSPLY_STRING CHAR(32) NOT NULL, KWD_ADD_USER CHAR(8) NOT NULL, KWD_ADD_TMDT TIMESTAMP NOT NULL, KWD_UPDT_USER CHAR(8) NOT NULL, KWD_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_NOTE	EXEC SQL DECLARE DBX_NOTE TABLE (NTE_Z TIMESTAMP NOT NULL, NTE_TTL CHAR(50) NOT NULL, NTE_QUAL CHAR(10) NOT NULL, NTE_DSPLY_TTL CHAR(50) NOT NULL, GRP_ID CHAR(10) NOT NULL, NTE_TYPE CHAR(1) NOT NULL, NTE_SHR_RSTRCT_IND CHAR(1) NOT NULL, TOP_Z TIMESTAMP, NTE_ADD_USER CHAR(8) NOT NULL, NTE_ADD_TMDT TIMESTAMP NOT NULL, NTE_UPDT_USER CHAR(8) NOT NULL, NTE_UPDT_TMDT TIMESTAMP NOT NULL, NTE_PRTY_VALUE CHAR(1) NOT NULL, NTE_TXT_ADD_USER CHAR(8) NOT NULL, NTE_TXT_ADD_TMDT TIMESTAMP NOT NULL, NTE_TXT_UPDT_USER CHAR(8) NOT NULL, NTE_TXT_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_TEXT_ATTR	EXEC SQL DECLARE DBX_TEXT_ATTR TABLE (TXX_Z TIMESTAMP NOT NULL, TXX_COMPLETE_IND CHAR(1) NOT NULL, TXX_ACTIVE_IND CHAR(1) NOT NULL, TXX_FMT_TYPE CHAR(1) NOT NULL, TXX_LN_SIZE SMALLINT NOT NULL, TXX_PG_SIZE SMALLINT NOT NULL, TXX_BLK_CNT SMALLINT NOT NULL, TXX_LN_CNT SMALLINT NOT NULL, TXX_ADD_USER CHAR(8) NOT NULL, TXX_ADD_TMDT TIMESTAMP NOT NULL, TXX_UPDT_USER CHAR(8) NOT NULL, TXX_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_TEXT_AUD	EXEC SQL DECLARE DBX_TEXT_AUD TABLE (TOP_Z TIMESTAMP NOT NULL, TXA_Z_TYPE CHAR(3) NOT NULL, TXA_ADD_USER CHAR(8) NOT NULL, TXA_ADD_TMDT TIMESTAMP NOT NULL,

Table	Declaration
	TXA_UPDT_USER CHAR(8) NOT NULL, TXA_UPDT_TMDT TIMESTAMP NOT NULL, TXX_CURR_Z TIMESTAMP, TXX_ACTV_Z TIMESTAMP) END-EXEC.
DBX_TEXT_BLK	EXEC SQL DECLARE DBX_TEXT_BLK TABLE (TXX_Z TIMESTAMP NOT NULL, TXB_BLK_SEQ SMALLINT NOT NULL, TXB_BGN_LN_SEQ SMALLINT NOT NULL, TXB_BGN_LN_OFFSET SMALLINT NOT NULL, TXB_LN_CNT SMALLINT NOT NULL, TXB_FREE_CNT SMALLINT NOT NULL, TXB_FREE_OFFSET SMALLINT NOT NULL, TXB_TEXT LONG VARCHAR NOT NULL) END-EXEC.
DBX_TEXT_TAG	EXEC SQL DECLARE DBX_TEXT_TAG TABLE (TXX_Z TIMESTAMP NOT NULL, TAG_ID CHAR(2) NOT NULL, TOP_Z TIMESTAMP NOT NULL, TAG_Z_TYPE CHAR(3) NOT NULL) END-EXEC.
DBX_TOP_KWD_J	EXEC SQL DECLARE DBX_TOP_KWD_J TABLE (TOP_Z TIMESTAMP NOT NULL, TKJ_Z_TYPE CHAR(3) NOT NULL, KWD_Z TIMESTAMP NOT NULL, TKJ_ADD_USER CHAR(8) NOT NULL, TKJ_ADD_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_TOP_REL_J	EXEC SQL DECLARE DBX_TOP_REL_J TABLE (TOP_Z TIMESTAMP NOT NULL, TRJ_SUBTOP_SEQ SMALLINT NOT NULL, TOP_SUBTOP_Z TIMESTAMP NOT NULL, TRJ_ADD_USER CHAR(8) NOT NULL, TRJ_ADD_TMDT TIMESTAMP NOT NULL, TRJ_UPDT_USER CHAR(8) NOT NULL, TRJ_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_TOPIC	EXEC SQL DECLARE DBX_TOPIC TABLE (TOP_Z TIMESTAMP NOT NULL, TOP_TTL CHAR(50) NOT NULL, TOP_QUAL CHAR(10) NOT NULL, TOP_DSPLY_TTL CHAR(50) NOT NULL, GRP_ID CHAR(10) NOT NULL, TOP_READ_PROT_IND CHAR(1) NOT NULL, TOP_HRC_RSTRCT_IND CHAR(1) NOT NULL, TOP_TXT_RSTRCT_IND CHAR(1) NOT NULL, TOP_CNC_RSTRCT_IND CHAR(1) NOT NULL, TOP_TXT_Z TIMESTAMP, TOP_ADD_USER CHAR(8) NOT NULL, TOP_ADD_TMDT TIMESTAMP NOT NULL, TOP_UPDT_USER CHAR(8) NOT NULL,

Table	Declaration
	TOP_UPDT_TMDT TIMESTAMP NOT NULL, TOP_FORM_IND CHAR(1) NOT NULL, TOP_BOOK_IND CHAR(1) NOT NULL, TOP_TXT_ADD_USER CHAR(8) NOT NULL, TOP_TXT_ADD_TMDT TIMESTAMP NOT NULL, TOP_TXT_UPDT_USER CHAR(8) NOT NULL, TOP_TXT_UPDT_TMDT TIMESTAMP NOT NULL) END-EXEC.
DBX_TS_MESSAGES	EXEC SQL DECLARE DBX_TS_MESSAGES TABLE (TSM_MSG_ID CHAR(6) NOT NULL, TSM_MSG_SUFFIX CHAR(2) NOT NULL, TSM_MSG_SEVER CHAR(1) NOT NULL, TSM_MSG_TEXT CHAR(78) NOT NULL) END-EXEC.
DBX_USER	EXEC SQL DECLARE DBX_USER TABLE (USR_ID CHAR(8) NOT NULL, GRP_ID_DFLT CHAR(10), GRP_ID_PLIST CHAR(10), USR_PSSWD CHAR(8) NOT NULL, USR_TTL_SRCH_AUTH CHAR(1) NOT NULL, USR_TXT_SRCH_AUTH CHAR(1) NOT NULL, USR_QUAL_SRCH_AUTH CHAR(1) NOT NULL, USR_KWD_SRCH_AUTH CHAR(1) NOT NULL, USR_FIND_AUTH CHAR(1) NOT NULL, USR_PLIST_AUTH CHAR(1) NOT NULL, USR_DSPLY_QUAL_IND CHAR(1) NOT NULL, USR_PG_PROMPT_IND CHAR(1) NOT NULL, USR_LAST_ACC_TMDT TIMESTAMP, USR_ADD_USER CHAR(8) NOT NULL, USR_ADD_TMDT TIMESTAMP NOT NULL, USR_UPDT_USER CHAR(8) NOT NULL, USR_UPDT_TMDT TIMESTAMP NOT NULL, USR_HLP_MODE CHAR(1) NOT NULL, USR_NAME CHAR(25) NOT NULL, USR_NL_CODE CHAR(3) NOT NULL) END-EXEC.
DBX_USER_PRINT	EXEC SQL DECLARE DBX_USER_PRINT TABLE (USR_ID CHAR(8) NOT NULL, PRT_TOP_TYPE CHAR(1) NOT NULL, PRT_HRC_BRK_LVL SMALLINT NOT NULL, PRT_NTE_TYPE CHAR(1) NOT NULL, PRT_NTE_GRP_ID CHAR(10) NOT NULL, PRT_NTE_DATE CHAR(6) NOT NULL, PRT_TYPE CHAR(1) NOT NULL, PRT_DEST_ID CHAR(8) NOT NULL, PRT_PG_LINES SMALLINT NOT NULL, PRT_PG_WIDTH SMALLINT NOT NULL, PRT_FLOW_IND CHAR(1) NOT NULL, PRT_DSPLY_QUAL_IND CHAR(1) NOT NULL, PRT_CC_IND CHAR(1) NOT NULL,

Table	Declaration
	PRT_DJDE_IND CHAR(1) NOT NULL, PRT_DESC CHAR(25) NOT NULL, PRT_UPDT_TMDT TIMESTAMP NOT NULL, PRT_JCL_LN_CNT SMALLINT NOT NULL, PRT_JCL_UPDT_TMDT TIMESTAMP) END-EXEC.
DBX_VARFLD	EXEC SQL DECLARE DBX_VARFLD TABLE (FLD_ID CHAR(32) NOT NULL, FLD_LEN SMALLINT NOT NULL, FLD_TYPE CHAR(1) NOT NULL, FLD_REQ CHAR(1) NOT NULL, FLD_EDIT CHAR(1) NOT NULL, FLD_ATTR CHAR(1) NOT NULL, FLD_PROGRAM CHAR(8) NOT NULL, FLD_DEFAULT CHAR(50) NOT NULL) END-EXEC.

Appendix: Technical Reference Information

IAF001 Intercept Attach Facility Parameters
DBXUXIAF User Exit (COBOL)

```
05  IAF001-PARMS.
    10  IAF001-PARM-LENGTH      PIC S9(4)  VALUE +374 COMP.
    10  IAF001-EXIT-NAME        PIC X(8)   VALUE 'DBXUXIAF'.
    10  IAF001-RETURN-PARMS.
        15  IAF001-RTN-CODE     PIC S9(4)  VALUE +0    COMP.
          88  IAF001-NORM-RTN              VALUE +0.
          88  IAF001-NO-INTERCEPT          VALUE +8.
          88  IAF001-FAIL-W-DUMP           VALUE +12.
        15  IAF001-RTN-REASON   PIC X(4)   VALUE SPACES.
        15  IAF001-TYPE-ATTACH  PIC X(1)   VALUE SPACES.
          88  IAF001-XCTL                  VALUE 'X'.
          88  IAF001-START                 VALUE 'S'.
        15  IAF001-BACKDROP-IND PIC X(1)   VALUE SPACES.
          88  IAF001-SEND-BACKDROP         VALUE 'Y'.
          88  IAF001-DONT-SEND             VALUE 'N'.
        15  IAF001-ATTACH-PGM   PIC X(8)   VALUE SPACES.
        15  IAF001-MESSAGE      PIC X(78)  VALUE SPACES.
        15  IAF001-FILLER       PIC X(8)   VALUE LOW-VALUES.
    10  IAF001-INPUT-PARMS.
        15  IAF001-TRANID       PIC X(4)   VALUE SPACES.
        15  IAF001-CURSOR-ROW   PIC S9(4)  VALUE +0    COMP.
        15  IAF001-CURSOR-COL   PIC S9(4)  VALUE +0    COMP.
        15  IAF001-FLD-ROW      PIC S9(4)  VALUE +0    COMP.
        15  IAF001-FLD-COL      PIC S9(4)  VALUE +0    COMP.
        15  IAF001-FLD-LENGTH   PIC S9(4)  VALUE +0    COMP.
        15  IAF001-INT-LVL      PIC S9(4)  VALUE +0    COMP.
        15  IAF001-INT-QID      PIC X(8)   VALUE SPACES.
        15  IAF001-AID          PIC X(1)   VALUE SPACES.
        15  IAF001-FILLER       PIC X(7)   VALUE LOW-VALUES.
    10  IAF001-MODIFIABLE-PARMS.
        15  IAF001-USERID       PIC X(8)   VALUE SPACES.
        15  IAF001-APPLID       PIC X(8)   VALUE SPACES.
        15  IAF001-SCRNID       PIC X(8)   VALUE SPACES.
        15  IAF001-GROUP        PIC X(10)  VALUE SPACES.
        15  IAF001-FLD-VALUE    PIC X(50)  VALUE SPACES.
        15  IAF001-FLD-QUAL     PIC X(10)  VALUE SPACES.
        15  IAF001-FILLER       PIC X(8)   VALUE LOW-VALUES.
10  IAF001-USER-AREA.
        15  FILLER              PIC X(128) VALUE LOW-VALUES.
```

IAF001B Intercept Attach Facility Parameters DBXUXIAF (ALC)

```
I001PARM DS    0F   INTERCEPT PARAMETERS
I001PLEN DS    H PARM LENGTH --- S/B 374
I001RCID DC    CL8'DBXUXIAF' PARM RECORD ID
       SPACE
I001OP   DS    0H   RETURN PARAMETERS
I001RC   DS    H RETURN CODE, AS PER:
I001NORM         EQU 0      - NORMAL, GO WITH ATTACH
I001PASS         EQU 8      - BYPASS ATTACH
I001FAIL         EQU 12     - ABEND WITH DUMP
       SPACE
I001REAS         DS  CL4    DUMP ID FOR ABEND
       SPACE
I001TYPE DS    C TYPE OF ATTACH, AS PER:
I001XCTL         EQU C'X'   - XCTL TO PGM  = I001XPID
I001STRT         EQU C'S'   - START TRANID = I001STID
       SPACE
I001BUFF DS    C BUFFER DISPOSITION, AS PER:
I001SNDY         EQU C'Y'   - SEND AS PROTECTED BACKDROP
I001SNDN         EQU C'N'   - DON'T SEND BUFFER
       SPACE
I001ATID         DS  0CL8   ATTACH ID - PROGRAM OR TRANID
I001XPID         DS  0CL8   PROGRAM TO XCTL TO
I001STID         DS  CL4    TRANID TO START
       DS        CL4 FILLER
I001MSG          DS  CL78   INTERCEPTION REJECTION MESSAGE
       DS        CL8 FILLER
I001POL          EQU *-I002PARM   PARM LENGTH EXLUSIVE OF USER AREA
I001IP DS        0H  INPUT PARAMETERS
I001TRAN         DS  CL4    INPUT TRANID
I001CROW         DS  H      CURSOR ROW
I001CCOL         DS  H      CURSOR COLUMN
I001FROW         DS  H      FIELD ROW
I001FCOL         DS  H      FIELD COLUMN
I001FLEN         DS  H      FIELD LENGTH
I001ILVL         DS  H      INTERCEPT LEVEL
I001IQID         DS  CL8    INTERCEPT QID
I001AID          DS  C      INTERCEPT AID
       DS        XL7 RESERVED
       SPACE
I001UMF          DS  0H     USER-MODIFIABLE FIELDS
I001USER         DS  CL8    USERID (NULL ON ENTRY)
I001APPL         DS  CL8    APPLID
I001SCRN         DS  CL8    SCRNID
I001GRP          DS  CL10   GROUP
       SPACE
I001FVAL         DS  CL50   EXTRACTED FIELD VALUE
I001FQAL         DS  CL10   FIELD QUAL (NULL ON ENTRY)
       SPACE
       DS        CL8 FILLER
       SPACE     2
```

```
I001UXAR         DS  CL128  IAF USER AREA
       SPACE
I001PDL          EQU *-I001PARM    PAR LENGTH - S/B 374
```

IAF002 Parameters for Intercept Attach Facility Target Application (COBOL)

```
01  IAF002-PARMS.
    05  IAF002-LENGTH           PIC S9(4)  VALUE +314 COMP.
    05  IAF002-REC-NAME         PIC X(8)   VALUE 'DBA IAF'.
    05  IAF002-TRANID           PIC X(4)   VALUE SPACES.
    05  IAF002-CURSOR-ROW       PIC S9(4)  VALUE +0    COMP.
    05  IAF002-CURSOR-COL       PIC S9(4)  VALUE +0    COMP.
    05  IAF002-FLD-ROW          PIC S9(4)  VALUE +0    COMP.
    05  IAF002-FLD-COL          PIC S9(4)  VALUE +0    COMP.
    05  IAF002-FLD-LENGTH       PIC S9(4)  VALUE +0    COMP.
    05  IAF002-INT-LVL          PIC S9(4)  VALUE +0    COMP.
    05  IAF002-INT-QID          PIC X(8)   VALUE SPACES.
    05  IAF002-AID              PIC X(1)   VALUE SPACES.
    05  IAF002-RTN-TRANID       PIC X(4)   VALUE SPACES.
    05  IAF002-USERID           PIC X(8)   VALUE SPACES.
    05  IAF002-APPLID           PIC X(8)   VALUE SPACES.
    05  IAF002-GROUP            PIC X(10)  VALUE SPACES.
    05  IAF002-SCRNID           PIC X(8)   VALUE SPACES.
    05  IAF002-FLD-VALUE        PIC X(50)  VALUE SPACES.
    05  IAF002-FLD-QUAL         PIC X(10)  VALUE SPACES.
    05  IAF002-INT-FUNCTION     PIC X(1)   VALUE SPACES.
      88  IAF002-INT-NOTE                  VALUE 'N'.
      88  IAF002-INT-NOTE-LIST             VALUE 'L'.
      88  IAF002-INT-HELP                  VALUE 'H'.
      88  IAF002-INT-USER                  VALUE 'U'.
    05  IAF002-INT-TYPE         PIC X(1)   VALUE SPACES.
      88  IAF002-INT-GENERIC               VALUE 'G'.
      88  IAF002-INT-BMS                   VALUE 'B'.
    05  IAF002-TCTTE-EXTRACT.
        15  IAF002-TCTTE-FIB    PIC X(1)   VALUE LOW-VALUES.
        15  IAF002-TCTTE-TRIX   PIC X(1)   VALUE LOW-VALUES.
    05  IAF002-TERM-CASE        PIC X(1)   VALUE SPACES.
      88  IAF002-LOWER                     VALUE 'L'.
      88  IAF002-UPPER                     VALUE 'U'.
    05  IAF002-EXTDS-IND        PIC X(1)   VALUE LOW-VALUES.
      88  IAF002-NO-EXTDS                  VALUE 'N'.
      88  IAF002-EXTDS-POSSIBLE            VALUE 'Y'.
    05  IAF002-ALTSCRN-IND      PIC X(1)   VALUE LOW-VALUES.
      88  IAF002-SIZE-DEFAULT              VALUE 'N'.
      88  IAF002-SIZE-ALT                  VALUE 'Y'.
    05  IAF002-SCRN-COLS        PIC S9(4)  VALUE +0    COMP.
    05  IAF002-SCRN-LINES       PIC S9(4)  VALUE +0    COMP.
    05  IAF002-HOST-OLR-IND     PIC X(1)   VALUE SPACES.
      88  IAF002-HOST-OLR                  VALUE 'Y'.
    05  IAF002-ATI-IND          PIC X(1)   VALUE SPACES.
      88  IAF002-NO-START                  VALUE 'N'.
      88  IAF002-START-OK                  VALUE 'Y'.
    05  IAF002-FILL-A           PIC X(40)  VALUE LOW-VALUES.
    05  IAF002-IAF-PARMS        PIC X(128) VALUE LOW-VALUES.
```

IAF002B Parameters for Intercept Attach Facility Target Application (ALC)

```
I002PARM DS    0F   INTERCEPT PARAMETERS
I002PLEN DS    H PARM LENGTH --- S/B 314
I002RCID DC    CL8'DBA IAF'      PARM RECORD ID
      SPACE
I002TRAN         DS  CL4    INTERCEPTED TRANID
I002CROW         DS  H      INTERCEPTEDCURSOR ROW
I002CCOL         DS  H      INTERCEPTED CURSOR COLUMN
I002FROW         DS  H      INTERCEPTED FIELD ROW
I002FCOL         DS  H      INTERCEPTED FIELD COLUMN
I002FLDL         DS  H      INTERCEPTED FIELD LENGTH
I002ILVL         DS  H      INTERCEPT LEVEL
I002IQID         DS  CL8    INTERCEPT QID
I002AID          DS  C      INTERCEPTION AID
I002TXCD         DS  CL4    RETURN TRANID TO START
      SPACE
I002USER         DS  CL8    USERID - SET BY 'DBXUXIAF'
I002APPL         DS  CL8    APPLID
I002GRP          DS  CL10   GROUP
I002SCRN         DS  CL8    SCRNID
      SPACE
I002FVAL         DS  CL50   EXTRACTED FIELD VALUE
I002FQAL         DS  CL10   FIELD QUAL - SET BY 'DBXUXIAF'
      SPACE
I002FUNC DS    C INTERCEPT FUNCTION, AS PER:
I002HELP         EQU C'H'   - HELP
I002NOTE         EQU C'N'   - NOTE
I002LIST         EQU C'L'   - LIST
I002IAF          EQU C'U'   - INTERCEPT ATTACH
      SPACE
I002TYPE DS    C INTERCEPT TYPE, AS PER:
I002GEN          EQU C'G'   - GENERIC
I002BMS          EQU C'B'   - BMS
      SPACE
I002XTCT         DS  0CL2   TERMINAL FIB/TRIX
I002FIB          DS  C      TERMINAL FIB
I002TRIX         DS  C      TERMINAL TRIX
      SPACE
I002CASE DS    C TERMINAL CASE, AS PER:
I002LC EQU       C'L'       - CASE SET TO LOWER
I002UC EQU       C'U'       - CASE SET TO UPPER
      SPACE
I002EXT          DS    C    TERMINAL EXTDS CAPABLE, AS PER:
I002EXTN         EQU C'N'   - NOT CAPABLE
I002EXTY         EQU C'Y'   - CAPABLE
      SPACE
I002ALT          DS    C    TERMINAL IN ALTSCRN MODE, AS PER:
I002ALTN         EQU C'N'   - IN DEFAULT MODE
I002ALTY         EQU C'Y'   - IN ALTSCRN MODE
```

```
      SPACE
I002SSZC        DS  H      TERMINAL SCREENSIZE - COLUMNS
I002SSZR        DS  H      TERMINAL SCREENSIZE - ROWS
      SPACE
I002HOLR        DS    C    INTERCEPTED TRANID OLR, AS PER:
I002HOY         EQU C'Y'   - IT IS AN OLR TRANID
I002HON         EQU C' '   - IT ISN'T AN OLR TRANID
      SPACE
I002ATI         DS    C    TERMINAL CAN HANDLE START, AS PER:
I002ATIN        EQU C'N'   - NO
I002ATIY        EQU C'Y'   - YES
      SPACE
      DS        CL40       RESERVED
      SPACE
I002FDL         EQU *-I002PARM   FIX LENGTH - S/B 168
      SPACE
      DS        CL40       RESERVED
      SPACE
I002UXAR        DS  CL128  USER ATTACH FACILITY PARM AREA
      SPACE
I002PDL         EQU *-I002PARM   PARM LENGTH - S/B 314
      SPACE
```

OIB001 The OLR Interface Block for OLR Attach Facility (COBOL)

```
01  OIB-RECORD.
           05  OIB-REC-LENGTH      PIC S9(4)   VALUE +512  COMP.
           05  OIB-REC-ID          PIC X(8)    VALUE 'DBXOIB'.
*-------------------------------------------------------------*
*   RETURN INFORMATION FROM OLR                               *
*-------------------------------------------------------------*
           05  OIB-RETURN-PARMS.
               10  OIB-RTN-CODE    PIC S9(4)   VALUE +0    COMP.
                 88  OIB-NORM-RTN             VALUE +0.
                 88  OIB-KEY-FAILURE          VALUE +8.
                 88  OIB-FAILURE              VALUE +12.
               10  OIB-MESSAGE     PIC X(72)   VALUE SPACES.
               10  FILLER          PIC X(12)   VALUE SPACES.
*-------------------------------------------------------------*
*   INPUT  PARMS FROM USER PROGRAM                            *
*-------------------------------------------------------------*
           05  OIB-INPUT-PARMS.
               10  OIB-REQ-TYPE    PIC X(2)    VALUE SPACES.
                 88  OIB-ADMIN-MENU           VALUE 'A '.
                 88  OIB-REFERENCE            VALUE 'R '.
                 88  OIB-HELP-LIST            VALUE 'H '.
                 88  OIB-FIELD-HELP           VALUE 'HF'.
                 88  OIB-NOTEPAD              VALUE 'NP'.
                 88  OIB-NOTE-AUTO-ADD        VALUE 'NA'.
                 88  OIB-NOTE-LIST            VALUE 'NL'.
                 88  OIB-NOTE-READ            VALUE 'NR'.
                 88  OIB-NOTE-UPDATE          VALUE 'NU'.
               10  OIB-RTN-MODE    PIC X       VALUE SPACE.
                 88  OIB-RTN-XCTL             VALUE 'X'.
                 88  OIB-RTN-START            VALUE 'S'.
               10  OIB-RTN-PGM     PIC X(8)    VALUE SPACES.
               10  OIB-EXTDS-IND   PIC X       VALUE SPACES.
                 88  OIB-NO-EXTDS             VALUE 'N'.
                 88  OIB-EXTDS                VALUE 'Y'.
               10  OIB-ALTSCRN-IND PIC X       VALUE SPACES.
                 88  OIB-NO-ALTSCRN           VALUE 'N'.
                 88  OIB-ALTSCRN              VALUE 'Y'.
               10  OIB-BYPASS-KEYFAIL-MSG-IND
                                   PIC X       VALUE SPACES.
                 88  OIB-SEND-KEYFAIL-MSG     VALUE 'N'.
                 88  OIB-BYPASS-KEYFAIL-MSG   VALUE 'Y'.
               10  OIB-RESTORE-SCRN-IND
                                   PIC X       VALUE SPACES.
                 88  OIB-NO-SCRN-RESTORE      VALUE 'N'.
                 88  OIB-RESTORE-SCRN         VALUE 'Y'.
               10  FILLER          PIC X       VALUE LOW-VALUES.
               10  OIB-TOP-GRP-ID  PIC X(10)   VALUE SPACES.
               10  OIB-NTE-GRP-ID  PIC X(10)   VALUE SPACES.
               10  OIB-APPL-ID     PIC X(8)    VALUE SPACES.
               10  OIB-SCRN-ID     PIC X(8)    VALUE SPACES.
               10  OIB-CURSOR-POS  PIC S9(4)   VALUE +0    COMP.
```

```
               10  OIB-USERID      PIC X(8)    VALUE SPACES.
               10  OIB-TOP-TTL     PIC X(50)   VALUE SPACES.
               10  OIB-TOP-QUAL    PIC X(10)   VALUE SPACES.
               10  OIB-NTE-TTL     PIC X(50)   VALUE SPACES.
               10  OIB-NTE-QUAL    PIC X(10)   VALUE SPACES.
               10  OIB-FF-TTL      PIC X(50)   VALUE SPACES.
               10  OIB-FF-QUAL     PIC X(10)   VALUE SPACES.
               10  OIB-VFD-QID     PIC X(8)    VALUE SPACES.
               10  OIB-VFD-COUNT   PIC S9(4)   VALUE +0    COMP.
               10  OIB-FORM-IND    PIC X       VALUE SPACES.
                 88  OIB-FORM                  VALUE 'Y'.
               10  FILLER          PIC X(161)  VALUE LOW-VALUES.
*----------------------------------------------------------------*
*   USER AREA - CONCATENATE USER AREA DEFINITION                 *
*----------------------------------------------------------------*
           05  OIB-USER-AREA.
               10  OIB-USER-AREA-LEN
                                   PIC S9(4)   VALUE +0    COMP.
```

OIB001B The OLR Interface Block for OLR Attach Facility (ALC)

```
I001PARM DS    0F                  PARAMETER LIST BEGINNING
I001LL   DS    H                   LIST LENGTH
         SPACE
I001BLKN DC    CL8'DBXOIB'         BLOCK ID
         SPACE 2
*-------------------------------------------------------------------
*   RETURN PARAMETERS
*
*-------------------------------------------------------------------
         SPACE 3
I001RPRM DS    0H                  RETURNED PARAMETERS
         SPACE
I001RTCD DS    H                   RETURN CODE, AS PER:
I001NORM EQU   0                   - NORMAL RETURN
I001KEY  EQU   8                   - KEY FAILURE
I001FAIL EQU   12                  - FAILURE
         SPACE
I001MSG  DS    CL72                MESSAGE TEXT RETURNED
         SPACE
         DS    CL12                RESERVED
         SPACE
I001RPDL EQU   *-I001RPRM          RETURNED PARMS MAX LENGTH
         SPACE 3
*-------------------------------------------------------------------
*
*   INPUT PARAMETERS
*
*-------------------------------------------------------------------
         SPACE 3
I001IPRM DS    0H                  INPUT PARAMETERS
         SPACE
I001REQT DS    XL2                 REQUEST TYPE, AS PER:
I001MENU EQU   C'A '               - ADMIN MENU
I001REF  EQU   C'R '               - REFERENCE
I001HLST EQU   C'H '               - HELP LIST
I001FHLP EQU   C'HF'               - FIELD HELP
I001NPAD EQU   C'NP'               - NOTEPAD
I001NOAA EQU   C'NA'               - NOTE AUTO-ADD
I001NLST EQU   C'NL'               - NOTE LIST
I001NORD EQU   C'NR'               - NOTE READ
I001NOUP EQU   C'NU'               - NOTE UPDATE
         SPACE
I001RMOD DS    X                   RETURN MODE, AS PER:
I001XCTL EQU   C'X'                - XCTL
I001STRT EQU   C'S'                - START
         SPACE
I001RTID DS    0CL8                RETURN ID (PROGRAM OR TRANID)
I001RPGM DS    0CL8                RETURN PROGRAM, FOR XCTL
I001RTRN DS    CL4                 RETURN TRANID, FOR IC START
         DS    CL4                 UNUSED
```

```
         SPACE
I001EXTD DS    C                   EXTENDED DATASTREAM, AS PER:
I001EXTY EQU   C'Y'                - HAVE EXTDS
I001EXTN EQU   C'N'                - NO EXTDS
         SPACE
I001ALT  DS    C                 USER TRANSACTION ALTSCRN, AS PER:
I001ALTY EQU   C'Y'                - YES IT IS
I001ALTN EQU   C'N'                - NOT ALTSCRN
         SPACE
I001BMSG DS    C                 BYPASS KEY-FAILURE MSG, AS PER:
I001BYPM EQU   C'Y'              - BYPASS KEY-FAIL MESSAGE WINDOW
I001SENM EQU   C'N'              - SEND KEY-FAIL MESSAGE WINDOW
         SPACE
I001RSTR DS    C                   RESTORE SCREEN, AS PER:
I001RESY EQU   C'Y'                - YES, RESTORE SCREEN
I001RESN EQU   C'N'                - NO, DON'T RESTORE SCREEN
         SPACE
         DS    CL1                 RESERVED
         SPACE
I001TGRP DS    CL10                GROUP TO ADD TOPICS TO
I001NGRP DS    CL10                GROUP TO ADD NOTES TO
         SPACE
I001APPL DS    CL8                 HELP REQUEST APPLID
I001SCRN DS    CL8                 HELP REQUEST SCREENID
I001CPOS DS    H                  0-REL CURSOR POSITION FOR HELP
         SPACE
I001USER DS    CL8                 USERID
         SPACE
I001TTTL DS    CL50                TOPIC TITLE
I001TQ   DS    CL10                TOPIC QUAL
         SPACE
I001NTTL DS    CL50                NOTE TOPIC TITLE
I001NTQ  DS    CL10                NOTE TOPIC QUAL
         SPACE
I001FFNM DS    CL50                FORM NAME
I001FFQ  DS    CL10                FORM QUAL
         SPACE
I001VFID DS    CL8                 VFD QID
I001VF#  DS    H                   NUMBER OF VFD'S IN QUEUE
         SPACE
I001FP   DS    X                   FORM PROVIDED IND, AS PER:
I001FPY  EQU   C'Y'                - YES, FORM IS PROVIDED
         SPACE
         DS    CL161               RESERVED
         SPACE
I001UALL DS    H                   USER AREA LENGTH
I001UAR  DS    0C                  BEGINNING OF USER AREA
         SPACE
I001IPDL EQU   *-I001PARM           INPUT PARM TOTAL LENGTH
```