

The OLR System[®]

Release 3.0

Customization Guide



Information in this document is subject to change without notice and does not represent a commitment on the part of DBA Software. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy this document in any form, regardless of reproduction medium, for any purpose other than the purchasing site's own use.

© 1996 DBA Software. All rights reserved.

06/06/97

Contents

Preface	About This Guide..... i
----------------	--------------------------------

Intended Audience.....	ii
Related Publications.....	ii
Chapter 1 Customizing the OLR System	1
Overview	2
Chapter 2 Setting System Parameters	5
System Parameters	6
Changing the Sysparm settings.....	11
Loading a new Sysparm table	11
Chapter 3 Print Services.....	13
OLR Online Print Facility	14
OLR Spool Print Facility	15
Chapter 4 Defining the OLR Intercept Service	17
OLR CICS Intercept Table.....	18
OLR CICS Intercept DDname Table.....	21
Chapter 5 Multiple Region (MRO) Configuration	23
Multiple Region (MRO) Configurations	24
Chapter 6 Overriding OLR TRANIDs	27
Overriding OLR Tranids.....	28
Chapter 7 OLR QMF Reports	29
Importing OLR QMF Reports.....	30
Running a QMF Report	32
Chapter 8 Defining Languages	33
Overview	34
Defining a Language	36
Creating Language Tables	37
Line Command Table (DBXORA).....	38
Command Table (DBXORC)	39
Literal Table (DBXORL).....	40
Message Table (DBXORM).....	40
Case Translation Table (DBXORX).....	41
The OLR Anchor Table (DBXANCHR)	43
Installing a Language	44
Chapter 9 Working with External Topics	45
Overview	46
External Topic program parameters	53
Chapter 10 OLR System Exits	59
The Intercept Exit (DBXUXINT).....	60
The OLR System Exit (DBXUXOLR).....	63
The User Profile Exit (DBXUXUSR)	65
The Note Form Selection Exit (DBXUXNFF):	67
Note Form Field Validation Exits	69
Using DBXEXTDS - Data Stream Extract Utility	71

Preface About This Guide

This guide is for system administrators and application developers who will customize the operation of the OLR System.

Chapter 1 provides an Overview of available customization features.

Chapter 2 describes the System Parameter settings you can use to tailor the user interface of the OLR System at your site.

Chapter 3 describes the available printing services and how to install and use them.

Chapter 4 describes the OLR Intercept Service and its installation and use.

Chapter 5 describes how to configure the OLR System to run in an MRO configuration.

Chapter 6 describes the changes you need to make if you want to override the OLR TRANIDs.

Chapter 7 describes the installation of the QMF Reports provided with the OLR System.

Chapter 8 describes the Language definition and customization features.

Chapter 9 explains how to use external topics to bring information into a topic from an external file, database or data source.

Chapter 10 describes the exits available for tailoring OLR System behavior as it is being used.

Intended Audience

This guide is intended as a reference for administrators who will be managing access to and directing the use of the OLR System to effectively meet your site's requirements.

Related Publications

OLR System Installation Guide

This guide provides complete instructions for the installation of the OLR System, OnLine Help, OnLine Reference, OnLine Notepad and the OLR API.

OLR System Guided Tours

The Guided Tours provide step by step tutorials for OnLine Help, OnLine Reference and OnLine Notepad. Use the Guided Tours to learn how to use each component of the OLR System.

OLR System User Guide

The User Guide describes the online features of OnLine Reference, OnLine Help, and OnLine Notepad and provides a general reference to each screen, processing option and command.

OLR System Import/Export Guide

The Import/Export Guide describes the facilities available for moving text between external word processors or personal computers and the OLR System.

OLR System Messages

The OLR System Messages Guide lists and describes all messages issued by the OLR System.

OLR API User Guide

The OLR API User Guide describes the application programming interface and its use. The OLR API is an optional feature of the OLR System.

Chapter 1 Customizing the OLR System

This chapter provides an overview of the customization features available with OLR System. It includes the following topics:

- Overview of Customization Features
- System Parameters
- MRO configuration
- QMF Reports
- Language Definition and Customization
- Using External Topics
- OLR System Exits.

Overview

The OLR System runs in a CICS configuration, using a DB2 database to manage its documents and related information. The installation cartridge provides a number of facilities you can use to customize the behavior of the OLR System at your site.

System Parameters table

You can alter the standard behavior of the OLR System by changing settings in the System Parameters table. This table is assembled into a load module and becomes resident when the OLR System is active.

When you install the OLR System, you provide an authorization password for your CPU as a System Parameter setting. You also declare the installed features and assign System Administrators.

You can use System Parameter settings to:

- limit the size of topics
- control the use of upper and lower case terminals
- designate the source of the Userid used by the OLR System
- activate MRO support
- activate alternate language support
- activate printing services
- control the use of search facilities
- activate system exits.

CICS Resources

The OLR System includes extensive support for the use of CICS services. You can use these resources to support:

- online and spool printing
- multiple region configurations based on CICS MRO support
- configuration and management of the OLR Intercept Service to enable hotkeys for OnLine Help and OnLine Notepad
- activation and use of a datastream trace facility.

QMF Reports

The OLR System provides a basic set of QMF reports for managing the information you store in the OLR Server. You can install these reports and customize or add to them. The OLR Server is a standard DB2 database, so you can also use other management and reporting tools with it.

Language Definition and Customization

You can use the Language support built into the OLR System to support the use of different languages in different parts of the world, or on different applications. You can also customize the English language elements of the OLR System interface.

External Topics

Some applications call for delivery of constantly changing information to the desktop. Others call for information currently available in another file, database or data source. You can use the External Topics support in OLR to dynamically create topics as the user requests them.

OLR System Exits

If your application needs to determine which topic to present, or which library to provide access to at the time the user requests information, you can use the OLR exits to examine requests and change the information OLR uses to access information.

You can use exits to alter the language which will be used by commands, messages and screen literals.

You can also use exits to maintain access logs.

Chapter 2 Setting System Parameters

The System Parameters Table provides settings you can use to control the behavior of the OLR System.

This chapter describes the System Parameter settings, explains how to change them, and how to install a new System Parameters table.

System Parameters

The System Parameters and their settings are described below. New parameters are flagged with a change bar. Default settings are underlined.

ORSYSPRM TYPE=INITIAL

TYPE=INITIAL contains general system parameters.

The PASSWORD, EXPDT, and FEATURES parameters are used during system start-up, and are required.

The SYSADM and DFLTUSER parameters are required.

Parameter	Description
PASSWORD	provide your CPU authorization code.
EXPDT	if this is a trial installation, provide the expiration date.
FEATURES	provide the feature list. OLH OnLine Help OLN OnLine Notepad OLR OnLine Reference API The OLR API
DFLTUSER	defines the userid to be passed to OLR for audit and authorization checking purposes for users who have not signed on to CICS. <i>The following parameters are optional.</i>
ALTLANG	YES/NO indicates if your site will customize the OLR System or use a different language using the OLR Language Support.
ALTSCRN	YES/NO indicates that alternate screen sizes are in use (i.e. any screen size other than 80x24).
APPONLY	YES/NO indicates whether notes can be updated or not. Setting this option to YES disables the writer's ability to update a note.
AUDID	determines what value the OLR System will use to identify a "USER" for audit and authorization checking purposes. USERID USERID from signon table will be used TERMINAL TERMINAL will be used OPID OPID from TCT will be used
CASE	this will set the default value for the terminal case mode within the OLR System. I Case Change inhibited (defaults to case mode of terminal) This is the required setting for CICS/ESA Release 3.2. U All input will be translated to Upper case. L Allows Upper and Lower case characters.
CCDEF	ORCC specifies the OLR Case Change transaction.
COMPRESS	YES/NO indicates whether the print data is to be stored in compressed format to save temporary storage queue space.

Parameter	Description
DATSRCH	<u>YES/NO</u> indicates whether the Author/Date Search facility is enabled on your OLR System.
DESTTYPE	determines the type of print destination to be used. <u>START</u> Print output will be sent directly to a printer <u>TDQ</u> Print output will be written to a transient data destination
DISPATCH	<u>ORXX</u> specifies the OLR Program Control transaction.
EXTDS	<u>YES/NO</u> indicates whether the extended data stream support is active in CICS.
EXTFIND	<u>YES/NO</u> indicates whether the Extended Find facility is enabled on your OLR System.
FLDCOLOR	<u>YELLOW</u> sets the color attribute for fields in OnLine Notepad forms.
FORMGRP	<u>FORMGRP</u> sets the name of the Group library which contains blank forms.
KWDSRCH	<u>YES/NO</u> indicates whether the Keyword Search facility is enabled on your OLR System.
MAXTEXT	<u>1200</u> You can limit the number of text lines for a topic. Provide the maximum number of lines.
MRO	<u>YES/NO</u> indicates whether the OLR System will be running in an MRO environment
NOHLPBP	<u>YES/NO</u> indicates whether a help reader will see a message when there is no help defined for an application, screen or field. If YES, the message is not sent to the screen.
OPTLIT	<u>YES/NO</u> indicates whether the 'F6=Options' literal will appear in pop-up Help Windows.
PRINT	<u>YES/NO</u> indicates whether online printing will be available within the OLR System.
PRINTDRV	<u>ORPD</u> specifies the tranid of the OLR Online Print Facility's Print Driver or the tranid of your spool print facility.
PRTYLIST	<u>YES/NO</u> activates note priority settings. When these are on, notes will appear in note lists by priority, then date.
PRTYPE	determines the type of print facility to be used. <u>CICS</u> OLR Online Print Facility will be used <u>SPOOL</u> a user site-provided spool print facility will be used
SCRNHELP	<u>YES/NO</u> indicates whether OnLine Help will default to Screen Level Help (if available) when a user requests help for a field where no Field Level Help has been defined.
SCPOS1	<u>YES/NO</u> sets the sticky cursor to always select the value starting at column 1 of the line holding the cursor.
SCTAB	<u>YES/NO</u> restricts the sticky cursor feature to work only when the cursor is on a table field in a help window. When YES, the sticky cursor is only active on table entries.
SPOOLDEST	specifies the destination ID that will be used by spool print facility. This only applies if PRTYPE=SPOOL.
TAGCOLOR	<u>RED</u> sets the color attribute for displaying hypertext pop-up links.

Parameter	Description
TTLSRCH	YES/NO indicates whether the Title Search facility is enabled on your OLR System.
ULPFIx	%% a two character External Topic marker
ULPLIM	256 User Text Exit load limit - sets a system limit on the number of lines of text that can be retrieved using a User Text Program.
UXINT	YES/NO indicates whether an Intercept User Exit program is to be invoked with OnLine Help and OnLine Notepad.
UXOLR	YES/NO indicates whether an OLR User Exit program is to be invoked upon stand-alone entry to the OLR System.
UXUSR	YES/NO indicates whether a User Profile User Exit program is to be invoked to override registered user profiles.

ORSYSPRM TYPE=SYSADM

The SYSADM entry is used to authorize a userid to use the System Administrator functions.

You can register one or more System Administrators using this statement. Include one statement for each System Administrator.

At least one System Administrator is required.

Parameter	Description
SYSADM	specifies an userid of a system administrator.

ORSYSPRM TYPE=LANGUAGE

The Language type entry is used with the OLR System's Language Support (NLS). Use the Language type entry if you want to customize the OLR System user interface to satisfy your site's language needs and preferences.

To support multiple languages, create a Language type entry for each language.

To change the default language elements used in the OLR System, provide a Language table for the language code SYS.

If these do not apply to you, you can ignore this type entry.

Parameter	Description
LANGUAGE	specifies the Language Code. XXX A 3 character language code. For example, FRA for French. SYS Indicates an alternate set of OLR Default Language elements.
SUFFIX	a two-character suffix label used to identify your site's language tables that will override any of the OLR Language Tables. These language tables store OLR commands, line action commands, messages, literals and character case translation.
FRAME	used to override the OLR default set of window characters. You can assign different top (FT), bottom (FB), left (FL), right (FR), upper left (UL), upper right (UR), lower left (LL) and lower right (LR) characters in hex format.
DESCRIPTION	15 character description for the Language code.
DATEFORM	specifies the date format. USA Displays dates in MM/DD/YY format. EUR Displays dates in DD.MM.YY format. ISO Displays dates in YY/MM/DD format.

ORSYSPRM TYPE=TRANID

TYPE=TRANID entries are typically used to enable OLR to be accessed from more than one TOR when OLR is installed in a single AOR in an MRO environment. Basically, for each TOR that will route requests to OLR, a separate set of OLR entry TRANIDs is required.

The TYPE=TRANID entry equates an entry TRANID to the Case Change and Print Driver TRANID that OLR must use to route Case Change and Print request to the invoking TOR.

TYPE=TRANID can also be used to override the TRANIDs that invoke OLR services.

If you are installing OLR in a single CICS region and you have no need to change any of the OLR TRANIDs, then TYPE=TRANID entries are not needed.

Four TYPE=TRANID entries make up a Tranid type entry set. These correspond to four transactions that access the OLR System:

OLRA	The OLR System Administration menu
OLRC	The OLR System Intercept entry TRANID
OLRX	The OnLine Reference Book List menu
OLHX	The OnLine Help Link List menu.

Each of these transactions can invoke two other transactions:

ORCC	Case Change transaction
ORPD	Print Driver transaction

The four parameters that are provided with each TYPE=TRANID are listed below.

Parameter	Description
ENTRAN	specifies the translated OLR entry transaction.
INTRAN	specifies the standard OLR System access transaction. <u>These values must match the OLR TRANIDs listed above.</u>
CCTRAN	Change this if you want to override ' ORCC '. This should be the same for all CCTRAN parameters in one Tranid type entry set. This will override the CCDEF parameter in the Initial macro entry.
PDTRAN	Change this if you want to override ' ORPD '. This should be the same for all PDTRAN parameters in one Tranid type entry set. This will override the PRINTDRV parameter in the Initial macro entry.

For example, if you want to override the default OLR TRANIDs, the TYPE=ENTRY set should look like:

```
ORSYSPRM TYPE=TRANID,ENTRAN='RLRX',INTRAN='OLRX',
          CCTRAN='RRC',PDTRAN='RRPD'
ORSYSPRM TYPE=TRANID,ENTRAN='RLHX',INTRAN='OLHX',
          CCTRAN='RRC',PDTRAN='RRPD'
ORSYSPRM TYPE=TRANID,ENTRAN='RLRA',INTRAN='OLRA',
          CCTRAN='RRC',PDTRAN='RRPD'
ORSYSPRM TYPE=TRANID,ENTRAN='RLRC',INTRAN='OLRC',
          CCTRAN='RRC',PDTRAN='RRPD'
```

ORSYSPRM TYPE=DESTID

The Destination ID type entry is used to define available printer destinations for the OLR Online Print Facility. For each available CICS printer at your site, assign a TYPE=DESTID.

If you provide a list of destination Ids, the OLR System will edit all print requests against the list. If you do not provide a list, print request destinations will not be validated.

Parameter	Description
DESTID	specifies an available CICS print destination ID of up to four characters.

ORSYSPRM TYPE=SPOOLID

The Spool ID type entry is used to define available spool IDs for a spool print facility used with the OLR System. For each available spool ID at your site, assign a TYPE=SPOOLID.

If you provide a list of destination Ids, the OLR System will edit all print requests against the list. If you do not provide a list, print request destinations will not be validated.

This is not required. If your site does not support spool print, skip this section.

Parameter	Description
SPOOLID	specifies an available spool ID of up to 8 characters.

Changing the Sysparm settings

To change the Sysparm settings:

1. Make changes to the **DBXORSIT** member in the **TABLES** dataset.
2. Tailor the **ASMORSIT** member in the **CNTL** dataset.
3. Run the job.

Upon completion, a new DBXORSIT load module will be created.

Loading a new Sysparm table

The Sysparm table is resident in the CICS region. The OLRA newcopy function must be used to load the new Sysparm table. The OLRA TRANID is only available to System Administrators.

To load a new Sysparm table:

1. Type the OLRA TRANID
2. Select the System Administration option
3. Select table DBXORSIT from the table list
4. Press the [F5] Newcopy key

The Sysparm table will be updated.

Chapter 3 Print Services

The OLR System provides support for printing to CICS and Spool print devices.

This chapter describes the print services and explains their setup.

OLR Online Print Facility

Overview

The OLR Online Print Facility allows a user to print topics or notes directly from the OLR System. The OLR Online Print Facility supports SCS and 3270 CICS printers, and sends output directly to a printer or indirectly via transient data destinations.

Setting up online printing

The following are guidelines in setting up parameters and destinations for online printing using the OLR Online Print Facility:

1. Define print parameter settings in the Initial type entry of the OLR System Parameters Table. Set these parameters as follows:

PRINT - must be set to 'YES'.

PRTYPE - must specify 'CICS'.

DESTTYPE - specify 'START' if you want the print output directly sent to a CICS printer. Or, specify 'TDQ' if you want the print output written to CICS transient data destinations.

COMPRESS - set to 'YES' if you want print data to be processed in compressed format.

2. Define print destinations to CICS.

If you want to send the print output directly to a CICS printer by specifying **DESTTYPE=START**, define these CICS printers to your Terminal Control Table. If these printers are not yet defined in CICS, add these printers as terminals to your TCT.

3. Define transient data queue destinations.

If you want to send print output via CICS transient data destinations by specifying **DESTTYPE=TDQ**, define these CICS transient data destinations to your Destination Control Table. For each DCT entry, use the OLR Print Driver tranid **ORPD** as the **TRANSID**. This transaction will automatically started when the DCT trigger level is reached.

4. Define valid destination Ids

You have the option to provide a list of available CICS destination Ids. If you provide a list, OLR will validate the user's requested destination ID. Use the **TYPE=DESTID** entry in the System Parameters table to define valid print destinations.

If **DESTTYPE=START**, **DESTID** would be a CICS printer defined as a four-character terminal ID in your CICS Terminal Control Table (TCT).

If **DESTTYPE=TDQ**, **DESTID** would be a four-character destination ID in your CICS Destination Control Table (DCT).

OLR Spool Print Facility

Overview

The OLR System formats the print data and allows provision for a spool print facility that may be available at your site. Instead of using the OLR Online Print Facility, your site may choose to use your own spool print facility.

For spool printing, the OLR System accommodates both ASA and non-ASA print formats.

For ASA print format, a special printing option 'Dynamic Job Description Entry' (DJDE) can be supported when printing outlines with text. If DJDE option is requested, each print line will contain

- an ASA carriage control character in the first position,
- a DJDE control character in the second position, and
- actual text starts in the third print position.

DJDE control characters can be used to assign different character fonts and sizes to each level of the outline and to the text in topics. OLR will insert the following DJDE characters:

1-5 the outline level, with the Book title or the top topic title as level 1, and each lower level numbered sequentially until level 5 is reached. If an outline has more than 5 levels, the levels beyond 5 will be assigned the DJDE code 5.

Space the text of a topic at any outline level

Your spool print facility can associate these DJDE control characters with specific font styles and sizes.

Setting up spool printing

1. Define print parameter settings in the Initial type entry of the OLR System Parameters Table. Edit these parameters as follows:

PRINT - must be set to 'YES'.

PRTYPE - must specify 'SPOOL'.

DESTTYPE - specify 'START' if you want OLR to directly invoke your spool print transaction. Or, specify 'TDQ' if you want OLR to indirectly send the print output to your spool print facility via transient data destination.

PRINTDRV - If DESTTYPE=START, this must specify the trandid of your site's spool print facility.

SPOOLDEST - If DESTTYPE=TDQ, this must specify the DCT destination ID used by your site's spool print facility.

COMPRESS - set to 'YES' if you want print data to be processed in compressed format.

2. You have the option to provide a list of available spool IDs. If you provide a list, OLR will validate the user's requested spool ID. Use TYPE=SPOOLID type entries in the System Parameters table to define valid spool IDs.

Chapter 4 Defining the OLR Intercept Service

The OLR Intercept Service is active at exit XZCATT in CICS. It provides a hot key intercept from CICS transactions to the OnLine Help and OnLine Notepad services.

If you have licensed the OLR API, you can also use the OLR Intercept Service to provide hot key activation of programs you develop.

This chapter describes the Intercept Service, its installation and use.

OLR CICS Intercept Table

The OLR CICS Intercept table controls the handling of hotkey functions for OnLine Help and OnLine Notepad by the OLR Intercept Service at XZCATT in your CICS region.

Use the procedures in this section to change the parameters used by this facility.

The **DBXORGTB** member in the **TABLES** dataset contains the default Intercept Parameter values.

OLRITAB TYPE=INITIAL

Parameter	Description
AUTAPP	DEMO specifies which application help topic will be used when an automatic intercept criteria is used with no application designated.
AUTGRP	DEMOGRP1 specifies the group library which will hold help topics when an automatic intercept criteria is used with no group library designated.
AUTSCRN	SCREEN specifies the screen help topic which will hold help topics when an automatic intercept criteria is used but no screen literal is found in the datastream.
ALTSCRN	YES/NO specifies whether alternate screen sizes are used (other than 80x24).
BMSX	YES/NO specifies whether the region is running CICS 4.1 with the BMS extension.
DETAIL	256 maximum number of details in the OLR CICS Intercept Table at enable time. The number of details corresponds to the sum of all the screens defined for all of the TRANID/PFKEY combinations. A plus (“+”) symbol will display beside the Active Detail value on the Intercept Status screen if the actual number of screens exceeds the DETAIL value. If this occurs, you should increase the setting of the detail parameter.
EXTDS	YES/NO indicates whether OnLine Help or OnLine Notepad will be used with applications that use extended data streams.
INTDD	OLRINT2 DDNAME of the OLR Intercept file in the CICS JCL
OFLOW	256 allocates the number of entries for any overflow from the PRIME area.
PFKXLATE	YES/NO if yes, PF keys 13-24 will automatically translate to 1-12.
PRIME	509 maximum number of entries in the OLR CICS Intercept Table at enable time. There is one ‘entry’ for each TRANID/PFKEY combination you want to intercept on. This must be a prime number.
SYMBI	YES/NO indicates whether any entries in the OLR Intercept File will be using Symbolic PF Keys to provide Help or Note support for terminals without function keys.
TERM	24 maximum number of terminals that can initiate intercepts at one time
TRACE	24 allocates the number of entries that can be viewed in the Intercept Trace Table accessed via the Intercept Status screen.
TRACEMSG	YES/NO indicates whether terminal buffer message contents for a terminal will be displayed in an Intercept Trace Table entry.

OLRITAB TYPE=SYSTEM

When OLR is installed, the Intercept Service is active in each TOR that will route requests to the AOR containing the OLR system. In addition, a subset of the Intercept Service is active in the OLR AOR as well.

It is a rule that each region containing Intercept Service components must have available an identical copy of the OLR Intercept Table. The purpose of this is twofold: Each TOR needs its own set of TRANIDs with which to invoke OLR, and also needs to advise OLR as to which TRANIDs must be used to return control to the invoking Intercept Service.

The TYPE=SYSTEM entry provides for this functionality. A TYPE=SYSTEM entry is coded for each system in which you expect to have active Intercept Service components.

When Intercept Service initialization is triggered in a given region, the SYSID in that region is used to locate the TYPE=SYSTEM entry that applies to it.

As a minimum, TYPE=TRANID is used to override default TRANIDs. Typically, it is required to support MRO configurations.

Single Region Configuration

If your site is not running in MRO and if you are not changing the TRANIDs of any OLR transactions, you need not provide a TYPE=SYSTEM entry. The defaults below will apply.

Multiple Region (MRO) Configuration

In MRO environments, assign a TYPE=SYSTEM entry for each TOR and AOR that will contain components of the OLR System.

Parameter	Description
SYSID	specifies the system identification (SYSID) of a CICS region that will run the OLR System or the OLR CICS Intercept.
SYSTYPE	specifies the type of CICS region. S/A stand-alone, non-MRO CICS region TOR terminal owning region AOR application owning region
SYNC	YES/NO applies only if SYSTYPE=TOR. This indicates whether the OLR CICS Intercept is to be synchronized in both TOR and AOR when it is re-initialized.
INITIAL	IS01 specifies the Intercept Initialization transaction.
SEQTERM	IS02 specifies the Intercept Sequential Terminal Initialization transaction
IENTRY	IS03 specifies the Intercept Entry transaction
RTRNDEF	IS05 Specifies the Intercept Return transaction for standard screen size terminals. This does not apply for the AOR. If you have multiple TORs, this must be unique in each TOR.
RTRNALT	IS07 Specifies the Intercept Return transaction for alternate screen size terminals. This does not apply for the AOR. If you have multiple TORs, this must be unique in each TOR.
SYNCH	IS08 specifies the Intercept Synchronization transaction
ABEND	IS0X specifies the Intercept Abend transaction

HOSTENT **OLRC** specifies the OLR Intercept entry TRANID.
STATUS **OLRS** specifies the Intercept Status screen transaction

Examples:

Standalone region	TOR	AOR
OLRITAB TYPE=SYSTEM, SYSID='CICS', SYSTYPE='S/A'	OLRITAB TYPE=SYSTEM, SYSID='TOR1', SYSTYPE='TOR', SYNC='YES',	OLRITAB TYPE=SYSTEM, SYSID='AOR1', SYSTYPE='AOR'

OLR CICS Intercept DDname Table

If you wish to use the OLRM transaction to maintain more than one Intercept file in a given CICS region, you may define the CICS DDname for each of these files in the Intercept DDname Table.(DBXPIDDN).

As shipped, the OLR DDname Table contains only one entry, for DDname **OLRINT2**. You may add additional entries to this table and re-assemble it.

This table is always loaded at entry to the OLRM Intercept File Maintenance transaction.

If the table contains only a single entry, then you will automatically go into maintenance on that file.

However, if the table contains more than one entry, you will be prompted to select one of the files prior to entry to the normal maintenance screen.

Use the procedures in this section to change the parameters used by this facility.

The **DBXPIDDN** member in the **TABLES** dataset contains the default Intercept File DDname value.

OLRIDDN TYPE=INITIAL

Parameter	Description
N/A	No parameters required.

OLRIDDN TYPE=DDNAME

Parameter	Description
DDNAME	8-byte DDname of file as defined to CICS .
DESC	description of the file at that DDname

OLRIDDN TYPE=FINAL

Parameter	Description
N/A	No parameters required.

OLR Intercept Initialization

The OLR intercept may be initialized manually via Tranid 'OLRS,' but is more often automatically initialized during CICS startup via the CICS PLT.

To initialize via PLT, you must include a PLT entry for program DBXIS010. This entry should be in the Stage 2 PLT. An entry is required in each TOR/ACR in which OLR is going to be active.

Chapter 5 Multiple Region (MRO) Configuration

As noted above in the chapters dealing with the Sysparm table and the Intercept Table, special capabilities have been provided to define multiple sets of TRANIDs for both OLR and the OLR Intercept Service.

These are intended, in the main, to facilitate the use of OLR in MRO environments.

This chapter discusses in some detail the considerations that must be addressed when setting up one of these environments for OLR.

Multiple Region (MRO) Configurations

Overview

The OLR System can be configured to operate in an MRO environment consisting of one or more TORs and a single OLR AOR.

If you are running CICS with CPSM (CICSplex System Manager/ESA), you may run with copies of OLR in multiple AOR's to balance your processing load.

- You can receive all terminal messages into one or more TORs, then route OLR transaction requests to an OLR AOR for processing. OLR requests can be OLR TRANIDs or messages for other TRANIDs that match OLR Intercept criteria.
- You can invoke the OLR System from more than one TOR to support multiple applications running in several other AORs. Generally speaking, multiple TORs can route to a single OLR AOR, but no TOR can route to more than one OLR AOR.

You create MRO configurations for the OLR System by defining local and remote resources to each CICS region. You then configure the OLR System to use these regions by updating the OLR System Parameters table and the OLR CICS Intercept table.

Using Terminal Owning Regions (TORs)

To route requests for the OLR System and the OLR CICS Intercept to a single AOR from one or more TORs.

For each TOR,

1. create a Tranid type entry set in the System Parameters Table,
2. create a System type entry in the OLR CICS Intercept Table.

When your CICS applications use OLR System hotkeys for OnLine Help or OnLine Notepad, the OLR System intercept can route these requests from the TOR directly to the AOR running the OLR System. When the user leaves the OLR System, it returns control back to the TOR using a TRANID you define in the OLR CICS Intercept table. This TRANID must also be registered to the TOR.

The following instructions show you how to define MRO configurations for the OLR System.

Defining a TOR in the System Parameters table

For each TOR, provide a set of Tranid type entries in the System Parameters Table.

- Each set includes four TYPE=TRANID entries.
- Each entry provides TRANID mappings for a transaction used to access the OLR System
- There are four transactions: **OLRX**, **OLHX**, **OLRA** and **OLRC**.
- There are four TRANIDs in each entry:
 - ENTRAN** is the remote name of the OLR transaction. This is unique for each TOR.
 - INTRAN** is the original TRANID of the OLR transaction.
 - CCTRAN** is the Case Change TRANID. This is unique for a TOR, but the same for all entries in a set.
 - PDTRAN** is the Print Driver TRANID. This can be the same for all TORs and all entries.

Example of Multiple TORs

Consider an MRO environment with two TORs. It would require a TRANID TYPE entry set for each TOR in the System Parameters table.

System Parameters Table Entries

TRANID TYPE ENTRY SET FOR TOR1	
ORSYSPRM	TYPE=TRANID,ENTRAN='OLR1',INTRAN='OLRX', CTRAN='ORC1',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLH1',INTRAN='OLHX', CCTRAN='ORC1',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLA1',INTRAN='OLRA', CCTRAN='ORC1',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLC1',INTRAN='OLRC', CCTRAN='ORC1',PDTRAN='ORP1'

TRANID TYPE ENTRY SET FOR TOR2	
ORSYSPRM	TYPE=TRANID,ENTRAN='OLR2',INTRAN='OLRX', CCTRAN='ORC2',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLH2',INTRAN='OLHX', CCTRAN='ORC2',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLA2',INTRAN='OLRA', CCTRAN='ORC2',PDTRAN='ORP1'
ORSYSPRM	TYPE=TRANID,ENTRAN='OLC2',INTRAN='OLRC', CCTRAN='ORC2',PDTRAN='ORP1'

CICS Intercept Table Entries

Each TOR would be defined in a SYSID entry in the OLR CICS Intercept table. The SYSID entry tells the OLR System which TRANID to return to when the user finishes using OnLine Help or OnLine Notepad.

The following parameters are used in a TYPE=SYSTEM entry in the OLR CICS Intercept Table:

SYSID	specifies the system identification of a TOR.
SYSTYPE	must specify 'TOR'.
SYNC	When you re-initialize the OLR Intercept Table in a TOR, the Intercept Table in the AOR needs to be re-initialized. When SYNC = YES, the OLR System automatically initiates the reinitialize in the AOR.
RTRNDEF	specifies the Intercept Return transaction for the default screen size. This must be unique for each TOR.
RTRNALT	specifies the Intercept Return transaction for alternate screen size. This must be unique for each TOR.

The two System type entries in the OLR CICS Intercept Table would look like the following:

TOR1	TOR2
OLRITAB TYPE=SYSTEM, SYSID='TOR1', SYSTYPE='TOR', SYNC='YES', RTRNDEF='I105', RTRNALT='I107'	OLRITAB TYPE=SYSTEM, SYSID='TOR2', SYSTYPE='TOR', SYNC='YES', RTRNDEF='I205', RTRNALT='I207'

PCT Entries

The transaction definitions used above need to be added to your PCT:

Tranid	TOR1 PCT definitions	TOR2 PCT Definitions	AOR PCT Definitions
OLRX	OLRX is remote with remote name OLR1	OLRX is remote with remote name OLR2	OLR1, OLR2 are local, pointing at DBXPR000
OLHX	OLHX is remote with remote name OLH1	OLHX is remote with remote name OLH2	OLH1, OLH2 are local, pointing at DBXPR000
OLRA	OLRA is remote with remote name OLA1	OLRA is remote with remote name OLA2	OLA1, OLA2 are local, pointing at DBXPR000
OLRC	OLRC is remote with remote name OLC1	OLRC is remote with remote name OLC2	OLC1, OLC2 are local, pointing at DBXPR000
ORCC	No change	No change	remote definitions for ORC1 and ORC2 with remote name ORCC
ORPD	No change	No change	remote definitions for ORP1 and ORP2 with remote name ORPD
IS03	IS03 is remote in AOR with remote name IS03	No change	IS03 is local, pointing at DBXPR000
IS05	No change	No change	remote definitions for I105 and I205 with remote name IS05
IS07	No change	No change	remote definitions for I107 and I207 with remote name IS07

CSM (CICSplex System Manager/ESA) Considerations

Note that if you are running with CPSM, and you wish CPSM to route OLR requests to more than one OLR AOR, you need only to leave the OLR tranids undefined in the TOR.

In this case CPSM will dynamically route to one of the AOR's in question.

Tranid IS03 is a special case however. It needs to be defined in the TOR as remote, but without the remote sysid included in the definition. CPSM will handle this as if it was not explicitly defined in the TOR, but OLR Intercept Service components that need the IS03 definition will be able to function normally.

Chapter 6 Overriding OLR TRANIDs

You can change the TRANIDs used by the OLR System to conform to your internal standards or to resolve conflicts with other applications running at your site.

This chapter describes the procedure for changing the TRANID assignments.

Overriding OLR Tranids

The OLR System Parameters Table (DBXORSIT) and the OLR CICS Intercept Table (DBXORGTB) contain most of the OLR System's tranids. You can change the TRANIDs by changing parameters in these tables.

The TRANID's and the associated parameter settings are shown below. If the TRANID is defined in the PCT or RCT, the respective column is marked 'YES'. When you change the TRANID, you must also update the PCT or RCT as appropriate.

Tranid	Table	Type entry	Parameter value	PCT	RCT
IS01	DBXORGTB	SYSTEM	INITIAL	YES	
IS02	DBXORGTB	SYSTEM	SEQTERM	YES	
IS03	DBXORGTB	SYSTEM	IENTRY	YES	YES
IS05	DBXORGTB	SYSTEM	RTRNDEF	YES	
IS07	DBXORGTB	SYSTEM	RTRNALT	YES	
IS08	DBXORGTB	SYSTEM	SYNCH	YES	
IS0X	DBXORGTB	SYSTEM	ABEND	YES	
OLRX	DBXORSIT	TRANID	ENTRAN	YES	YES
OLHX	DBXORSIT	TRANID	ENTRAN	YES	YES
OLRA	DBXORSIT	TRANID	ENTRAN	YES	YES
OLRC	DBXORGTB DBXORSIT	SYSTEM TRANID	HOSTENT ENTRAN	YES	
OLRS	DBXORGTB	SYSTEM	STATUS	YES	
ORCC	DBXORSIT	TRANID INITIAL	CCTRAN CCDEF	YES	
ORPD	DBXORSIT	TRANID INITIAL	PDTRAN PRINTDRV	YES	
ORXX	DBXORSIT	INITIAL	DISPATCH	YES	

To override a TRANID,

1. change the parameter settings in the System Parameters table or the OLR Intercept table,
2. change the PCT definitions in your CICS region
3. load the new SYSPARM table using the OLRA transaction, or
4. load the new OLR Intercept table using the OLRM transaction.

Chapter 7 OLR QMF Reports

The OLR System provides five QMF reports which can be used to retrieve information about the OLR System not currently available online:

- **Topic/Reference Link Report** - lists the subtopics of the requested topic, the topics it is a subtopic of, and its aliases. This was changed to accommodate longer OLR titles.
- **Topic/Help Link Report** - lists all of the Help Links connected to the requested topic. This was changed to accommodate longer OLR titles.
- **Topic/Note Link Report** - lists all of the Notes connected to the requested topic, in order by Group. This was changed to accommodate longer OLR titles.
- **Notes by Group Report** - lists all the Notes in the requested Group, in order by topic. This was changed to accommodate longer OLR titles.
- **Keyword Where-Linked Report** - lists all topics or notes linked to a requested keyword, in order by topic, note and group. This is new in this release of the OLR System.

The procedures in this chapter will describe how to import the procedures, forms and queries needed to run the QMF reports into your QMF environment, and how to produce the reports.

Importing OLR QMF Reports

Each OLR QMF report uses a procedure, a form and a query. Follow the steps listed below to import the procedures, queries and forms needed to run the OLR QMF Reports into your QMF environment. The upgraded procedures, queries and forms from this release should replace any existing ones.

- A. Edit the **QGRPNT**, **QTOPHLP**, **QTOPNTE**, **QTOPREF** and **QKWDOBJ** members in the **QMF** dataset as described below.

Qualify the table names in each query to correspond to the owner of the OLR/Server tables the reports will be run against. For example:

Change...

```
FROM DBX_NOTE           NTE
     ,DBX_TOPIC         TOP
     ,DBX_TOP_NOTE_J   TRJ
     ,DBX_GROUP        GRP
```

To...

```
FROM OWNERID.DBX_NOTE           NTE
     ,OWNERID.DBX_TOPIC         TOP
     ,OWNERID.DBX_TOP_NOTE_J   TRJ
     ,OWNERID.DBX_GROUP        GRP
```

- B. The procedures, queries and forms will be 'owned' on QMF by the user who actually performs the import to QMF. To make it possible for other users to run the reports, edit the **PGRPNT**, **PTOPHLP**, **PTOPNTE**, **PTOPREF** and **PKWDOBJ** members in the QMF dataset as described below.

Qualify the query and form names in each procedure to correspond with the TSO ID of the 'owner' of the queries and forms. For example:

Change...

```
RUN QUERY GRP_NTE_QUERY (FORM=GRP_NTE_FORM
```

To...

```
RUN QUERY INSTID.GRP_NTE_QUERY (FORM=INSTID.GRP_NTE_FORM
```

- C. Edit the **IMPORT** member in the **QMF** dataset as indicated in the annotated listing provided on page C-19. This member contains a procedure that will be imported to QMF to simplify the import of the OLR QMF reports.

At the QMF command prompt (in QMF) type:

```
IMPORT PROC IMPORT_PROC FROM 'YOUR.DBAOLR.QMF(IMPORT)' (S=Y
```

replacing 'YOUR.DBAOLR.QMF' with the DSN of the **QMF** dataset containing the **IMPORT** member. Press ENTER to import the procedure.

- D. This final step will complete the import of the procedures, queries and forms used by the OLR QMF reports.

At the QMF command prompt (in QMF) type:

```
RUN IMPORT_PROC
```

Press ENTER to run the procedure.

For a list of the procedures, queries and forms imported for each report, see the OLR QMF Report table on the following page.

Note: The query and procedure edits described in steps A and B above can be performed in QMF after the queries and procedures have been imported. In QMF, refer to the names the queries and procedures were imported under rather than the **QMF** dataset member names. See your IBM QMF documentation for instructions on displaying and saving queries and procedures in QMF.

Running a QMF Report

OLR QMF Reports

Report	Procedure	Query	Form
Topic/Reference Link	TOP_REF_PROC	TOP_REF_QUERY	TOP_REF_FORM
Topic/Help Link	TOP_HLP_PROC	TOP_HLP_QUERY	TOP_HLP_FORM
Topic/Note Link	TOP_NTE_PROC	TOP_NTE_QUERY	TOP_NTE_FORM
Notes by Group	GRP_NTE_PROC	GRP_NTE_QUERY	GRP_NTE_FORM
Keyword Where-Linked	KWD_OBJS_PROC	KWD_OBJS_QUERY	KWD_OBJS_FORM

The steps to generate an OLR QMF Report are detailed below. These instructions assume that the procedures, queries and forms needed to produce the reports have already been imported into QMF.

- A. Identify the procedure associated with the report you want to run by looking it up in the table above.

At the QMF command prompt (in QMF) type:

```
RUN OWNERID.TOP_REF_PROC
```

replacing 'OWNERID' with the TSO ID of the owner of the procedure (if you are the owner, you may omit the owner ID entirely), and replacing 'TOP_REF_PROC' with the name of the procedure associated with the report you are trying to produce. Press ENTER to run the procedure.

- B. QMF will prompt you to provide further information needed to generate the report. For the topic reports, the prompts will look like this:

```
&TOPTTL      ===>
&TOPQUAL     ===>
```

Provide the title and qualifier of the topic to be reported on in the fields provided on the prompt panel. The values you provide must be in upper case and enclosed in single quotes. If a value contains an embedded single quote, replace it with two single quotes. When prompted to provide a qualifier for a topic that has no qualifier, type a space enclosed in single quotes at the TOPQUAL prompt.

Press ENTER to generate the report.

The resulting report will be displayed on your QMF screen. After you log off TSO, a printed copy of the report will be routed to your SYSOUT queue.

Chapter 8 Defining Languages

This chapter describes the language customization features of the OLR System.

You can use the language features of the OLR System to support multiple languages, to change the primary interface language from English to another language, or to selectively change the language elements of the OLR System's English language interface.

This chapter explains the language definition and customization features, their use and implementation steps.

Overview

The OLR Language Support feature gives you the ability to tailor language elements such as commands, messages and screen literals to fit the needs of the people who will be using OLR at your site. You can select a different date format, or change the characters used to frame windows on OLR.

You can also use the OLR Language Support feature to define additional languages. This makes it possible to accommodate the needs of users who speak different languages, or have widely varying language preferences. Once you have defined a different language, the OLR System can be presented to different users in different ways.

A “language” in the OLR System consists of a complete set of language elements such as commands, line commands (used in the ACT column of the Edit Text and Change Outline screens), screen literals, messages, case translation tables (used to define lowercase characters and their uppercase translations), date format, and window frame characters.

Most of the elements mentioned are carried in load-module tables in the OLR System, collectively referred to as Language Tables. Other elements, such as the date format and window frame characters, are set when defining a language in the System Parameters Table.

The OLR System is shipped with an OLR default language for its basic user interface. The OLR default language uses English for its commands, messages and screen literals. It uses USA standards for date formats and provides a default set of window frame characters.

Setting up a Language

To use OLR Language Support, set the **ALTLANG** parameter in the System Parameters Table to ‘YES’.

When **ALTLANG = YES**, you can add languages to the OLR System by adding new **TYPE=LANGUAGE** entries to the System Parameters Table. Each language is identified by a three character language code.

For each language, you create a set of Language Tables.

Associating a Language with a User

Each registered OLR System user or generic user definition can be associated with a language code in its User Profile.

- If a Language code is not provided, the user will see the OLR default language.
- If a Language code is provided, it should correspond to the 3-character language code defined for a Language type entry in the System Parameters Table.

You can also write a User Profile User Exit program to dynamically assign a language to a user based on different criteria (e.g. transaction, application, screen, etc). For more information about the User Profile Exit, see the OLR API Guide.

The OLR Default Language

The OLR product is shipped with an OLR default language carried in a set of base Language Tables.

The OLR default language uses English for its commands, messages and screen literals. It uses USA standards for date formats and provides a default set of window frame characters.

You can change the English language specification or replace it with a different language by changing the characteristics set in the “SYS” Language Definition entry and its Language Tables.

OLR Language Elements

When you define a language, you can specify its:

- date format
- window frame characters
- commands
- screen literals
- messages
- case translation tables (lowercase to uppercase translations)

Most language elements listed are managed in load-module tables, collectively referred to as Language Tables. Date formats and window frame characters are specified in the System Parameters Table. The following table shows the language elements and where they are described.

Language Element	Location
date format	set using the DATEFORM parameter of a Language Definition entry in the System Parameters Table.
window frame characters	set using the FRAME parameter of a Language Definition entry in the System Parameters Table.
line commands	carried in the Line Command Table DBXORA
commands	carried in the Command Table DBXORC
screen literals	carried in the Literal Table DBXORL
OLR System messages (ORxxxx)	carried in the Message Table DBXORM
OLR Server messages (TSxxxx)	carried in the DB2 table DBX_TS_MESSAGES
case translation tables	carried in the Case Translation Table DBXORX

Defining a Language

Language Definition entries in the OLR System Parameters Table add languages to the OLR System. You create a Language Definition entry by adding a **TYPE=LANGUAGE** statement to the System Parameters table.

TYPE=LANGUAGE parameters

Parameter	Description																											
LANGUAGE	Required. Three character (alphanumeric) language code.																											
DESCRIPTION	Optional. Up to 15 character language description.																											
SUFFIX	Optional. Used to differentiate alternate versions of a Language Table. This parameter tells the OLR System which version of each Language Table to use with this language. For example: <p style="text-align: center;">SUFFIX=(ACT=01,CMD=01,LIT=,MSG=02,XLT=)</p> specifies version '01' of the Line Command (ACT) and Command (CMD) Tables, version '02' of the Message (MSG) Table, and the base versions of the Literal (LIT) and Case Translation (XLT) Tables. If this parameter is omitted, base versions will be used.																											
FRAME	Optional. Used to override the default window frame characters. The following shows how each window frame position is identified in the FRAME parameter: <table border="1" data-bbox="522 1213 922 1465"> <thead> <tr> <th>ID</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>FT</td> <td>6D</td> <td>top</td> </tr> <tr> <td>FB</td> <td>6D</td> <td>bottom</td> </tr> <tr> <td>FL</td> <td>4F</td> <td>left side</td> </tr> <tr> <td>FR</td> <td>4F</td> <td>right side</td> </tr> <tr> <td>UL</td> <td>4B</td> <td>upper left corner</td> </tr> <tr> <td>UR</td> <td>4B</td> <td>upper right corner</td> </tr> <tr> <td>LL</td> <td>7D</td> <td>lower left corner</td> </tr> <tr> <td>LR</td> <td>7D</td> <td>lower right corner</td> </tr> </tbody> </table> To alter a frame character, specify its ID and the new character (in hexadecimal format). For example: FRAME=(UL=6D,UR=6D,LL=6D,LR=6D) would use dashes in all four corners. Defaults would be used for all other positions.	ID	Default	Description	FT	6D	top	FB	6D	bottom	FL	4F	left side	FR	4F	right side	UL	4B	upper left corner	UR	4B	upper right corner	LL	7D	lower left corner	LR	7D	lower right corner
ID	Default	Description																										
FT	6D	top																										
FB	6D	bottom																										
FL	4F	left side																										
FR	4F	right side																										
UL	4B	upper left corner																										
UR	4B	upper right corner																										
LL	7D	lower left corner																										
LR	7D	lower right corner																										
DATEFORM	Optional. Sets the date format for a language. Options are: <table border="1" data-bbox="522 1675 1120 1755"> <tbody> <tr> <td>USA</td> <td>-</td> <td>USA standard:</td> <td>MM/DD/YY</td> </tr> <tr> <td>EUR</td> <td>-</td> <td>European standard:</td> <td>DD.MM.YY</td> </tr> <tr> <td>ISO</td> <td>-</td> <td>ISO:</td> <td>YY/MM/DD</td> </tr> </tbody> </table>	USA	-	USA standard:	MM/DD/YY	EUR	-	European standard:	DD.MM.YY	ISO	-	ISO:	YY/MM/DD															
USA	-	USA standard:	MM/DD/YY																									
EUR	-	European standard:	DD.MM.YY																									
ISO	-	ISO:	YY/MM/DD																									

A sample **TYPE=LANGUAGE** entry is as follows:

```
ORSYSPRM TYPE=LANGUAGE, LANGUAGE='SPA', DESCRIPTION='SPANISH',
SUFFIX=(LIT=02,CMD=01), DATEFORM='EUR',
FRAME=(LT=4B,FL=5C)
```

Creating Language Tables

There are five language tables:

Table ID	Description	Contents
DBXORA	Line Command Table	line commands used by text editor.
DBXORC	Command Table	commands and function key labels.
DBXORL	Literal Table	screen literals.
DBXORM	Message Table	messages
DBXORX	Case Translation Table	upper/lower case translation characters

The OLR default language is defined in base Language Tables with these names.

You can replace these tables to change elements of the standard user interface, or you can create new tables to specify elements of a language you defined.

When you create a new version of a language table, you add a 2-character suffix to its name to identify it.

For example, the base Command Table is called **DBXORC**. If you create another version of the table, you might assign it a suffix of '01'. The Command Table for the alternate language would then be called **DBXORC01**.

The Language Definition entry for the language would use the **SUFFIX** parameter with a **CMD=01** clause to assign the '01' suffixed version of the Command Table to that language.

A version of a Language Table can be shared by more than one language. Also, you can use different versions of tables when defining a language.

To create a new version of a language table:

1. Determine which Language Table you want to create.
2. Create a new copy of the source for the table. Add a suffix to the table name.
3. Edit the source for the new table.
4. Assemble and link the new table.
5. Add a PPT entry for the new table.
6. Add an entry in the OLR System Anchor Table (**DBXANCHR**) for the new Language Table.

Detailed instructions for modifying each table follow.

Line Command Table (DBXORA)

A copy of the source statements for the Line Command Table is available in the **DBXORA** member of the **TABLES** dataset on the installation/maintenance tape. Copy this member and rename it using a suffix.

The source statements begin with a **TYPE=INITIAL** statement, which establishes the table suffix:

```
OLRACT TYPE=INITIAL,SUFFIX='01'
```

The source statements end with a **TYPE=FINAL** statement.

```
OLRACT TYPE=FINAL
```

Between these, **TYPE=ALTERNATE** entries are used to redefine commands. You only need to include entries for the line commands you want to change.

TYPE=ALTERNATE parameters

Parameter	Description
BASEACT	This identifies the default value of the line command you want to override. <u>Do not</u> change this value.
ACT	Use this to define the line command you want to use. It can have up to two characters. It must be non blank and cannot be the same as any of the existing line commands.

For example, to change the Insert Bookmark command from **IB** to **BA**, add the following entry:

```
OLRACT TYPE=ALTERNATE,BASEACT='IB', ACT='BA'
```

Generating a new Line Command Table

To assemble and link a new Line Command Table, tailor the **ASMORA** member in the **CNTL** dataset. Run the job to create the new Line Command Table.

Command Table (DBXORC)

The Command Table specifies the OLR System commands and the command literals displayed for them when function keys or pull-down menus are available.

A copy of the source statements for the Command Table is available in the **DBXORC** member of the **TABLES** dataset on the installation/maintenance tape. Copy this member and rename it using a suffix.

The source statements begin with a **TYPE=INITIAL** statement, which establishes the table suffix:

```
OLRCMD TYPE=INITIAL,SUFFIX='01'
```

The source statements end with a **TYPE=FINAL** statement.

```
OLRCMD TYPE=FINAL
```

Between these, **TYPE=ALTERNATE** entries are used to redefine commands. You only need to include entries for the line commands you want to change.

TYPE=ALTERNATE parameters

Parameter	Description
BASECMD	This identifies the default value of the command you want to override. <u>Do not</u> change this value.
CMD	Use this to define the command you want to use. It can have up to eight characters, must be non-blank, and cannot be the same as any of the existing commands.
ALIAS	Use this to define the short form of the command you want to use. It can have up to eight characters, must be non-blank, and cannot be the same as any of the existing base or alternate commands, or any of the base or alternate aliases.

For example, to change the **ADDTOP** command to **NEWTOP**, with an alias of **NT**, include the following entry:

```
OLRCMD TYPE=ALTERNATE,BASECMD='ADDTOP',
      CMD='NEWTOP',ALIAS='NT'
```

Generating a new Command Table

To assemble and link a new Command Table, tailor the **ASMORC** member in the **CNTL** dataset. Run the job to create the new Command Table.

Literal Table (DBXORL)

The Literal Table contains the literals used in all screens and windows. These literals include screen titles, title labels, title bar headings, column headings, field labels, comments or instructions, message prompts, menu options, status literals, and others.

A copy of the source statements for the Line Command Table is available in the **DBXORL** member of the **TABLES** dataset on the installation/maintenance tape. Copy this member and rename it using a suffix.

When you create a Literal Table you need to build it in its entirety, including all of the statements provided in the base copy of the source.

The source statements begin with a **TYPE=INITIAL** statement, which establishes the table suffix:

```
OLRLIT TYPE=INITIAL, SUFFIX='01'
```

The source statements end with a **TYPE=FINAL** statement: **OLRLIT TYPE=FINAL**

Literal definitions on the Literal Table are grouped by screen, beginning with an initial section of generic literals. Generic literals are the general terms, words or phrases that are commonly used across the different OLR System screens and windows. Each field literal used on a given screen is grouped under the Screen ID (map name) for that screen.

Each **TYPE=SCREEN** statement establishes the purpose for which the following literals will be used. In the case of **GENERIC**, the literals are general purpose. In all other cases, the literals will be used in displays generated for the screen named.

TYPE=SCREEN entries must not be moved or modified.

Each **TYPE=LITERAL** defines a literal value that will be displayed in a particular field on the current screen or window.

TYPE=LITERAL parameters

Parameter	Description
LENGTH	This indicates the maximum number of characters for the literal value being defined. Do not change this value.
ID	This identifies a particular field on the screen or window. Do not change this value.
VALUE	Overtyping this value to change the literal to be displayed. The value must be enclosed in single quotes, and cannot have more characters than indicated in the LENGTH= parameter.

Assume the Literal Table you are modifying contains the following literal definition:

```
OLRLIT TYPE=LITERAL, LENGTH='10', ID='BOOKS', VALUE='BOOKS'
```

To change the literal “**BOOKS**” to “**Manuals**”, change the statement to:

```
OLRLIT TYPE=LITERAL, LENGTH='10', ID='BOOKS', VALUE='Manuals'
```

Generating a new Literal Table

To assemble and link a new Literal Table, tailor the **ASMLIT** member in the **CNTL** dataset. Run the job to create the new Literal Table.

Message Table (DBXORM)

The Message Table (**DBXORM**) contains the complete set of all OLR System messages.

A copy of the source statements for the Message Table is available in the **DBXORM** member of the **TABLES** dataset on the installation/maintenance tape. Copy this member and rename it using a suffix.

The source statements begin with a **TYPE=INITIAL** statement, which establishes the table suffix, and the special character used to indicate presence of dynamic variables in messages:

```
OLRMSG TYPE=INITIAL,SUFFIX='01',PARMSUB='$'
```

The source statements end with a **TYPE=FINAL** statement: **OLRMSG TYPE=FINAL**

Between these, **TYPE=MSG** entries are used to redefine commands. You only need to include entries for the line commands you want to change.

TYPE=MSG parameters

Parameter	Description
ID	This is the internal OLR ID of the message. <u>Do not</u> change this.
SEVERITY	This is the severity of the message. <u>Do not</u> change this.
TEXT	Type over this value to change the message to be displayed. The value must be enclosed in single quotes, and cannot have more than 70 characters. Any single quote appearing within the text must be coded as two single quotes to satisfy the IBM Assembler. OLR messages contain symbolic parameters to support dynamic value substitution at execution time. These parameters, in the base table, are "\$1" - "\$5". When you encounter one of these in the text of a message you are altering, you are free to move it to another location in the text. You are not free, however, to delete it.
PARMSUB	To set a different special character that the PARMSUB value set in the INITIAL statement, type a character in quotes here

Assume the base table has the following definition:

```
OLRMSG TYPE=MSG, ID=OR0104, TEXT='No books found for $1'
```

To change the text to "No \$20.00 books found for \$1", the result of your changes should be:

```
OLRMSG TYPE=MSG, ID=OR0104, TEXT='No $20.00 books found for %1',
PARMSUB='%'
```

A parm substitution character was specified so that the "\$" in the message text would not be interpreted as a parameter. This parm substitution character applies only to this message. The global "\$" applies unless explicitly overridden at the message level.

Generating a new Message Table

To assemble and link a new Message Table, tailor the **ASMOMSG** member in the **CNTL** dataset. Run the job to create the new Message Table.

Case Translation Table (DBXORX)

The OLR Case Translation Table (**DBXORX**) contains two 256-byte translate tables. These default translate tables are based on IBM standard U.S. bit pattern assignment.

The first table is the “**UPPER**” table. This table is used to translate characters considered to be lower case to their upper case equivalents. For the OLR base system the translation is:

“**a**” - “**z**” are translated to “**A**” - “**Z**”

All other characters are translated to themselves

The **UPPER** table is used to translate user-keyed input to upper case where applicable.

The second table is the “**UPLow**” table. This table is used to identify characters considered to be lower case. For the OLR base system the definition of lower case characters is:

“**a**” - “**z**” are defined as lowercase.

No other characters are defined in this table, as any non-null character included is by definition lower case.

This table is used to determine if a topic or note contains lower case characters to identify the need for case change processing or warnings.

A sample copy of the Case Translation Table (**DBXORX**) is in the **TABLES** dataset provided in the installation/maintenance tape.

To create a new Case Translation Table, copy this base table definition to a new definition, make the modifications you wish, then assemble it under a new name.

As for all of the tables described above, table definition begins with a **TYPE=INITIAL**, and is terminated with a **TYPE=FINAL**. Individual translation tables are defined via **TYPE=UPPER** and **TYPE=UPLow** entries. The base macro is **OLRXLT**.

If you create a table using only a **TYPE=INITIAL** and a **TYPE=FINAL** you would end up with an **UPPER** and **UPLow** with the base definitions noted above. I.E. these will both default.

To define either of these tables, you must code a **TYPE=** and specify all hex byte configurations for the table by specific range. To do this you must use the range parameters “**X00**” through “**XF0**”.

The basic rules for a range parameter are as follows:

- Each range parameter defines 16 bytes of the corresponding translate table, and must be 32-bytes in length. Only “**A**” - “**F**” and “**0**” to “**9**” may be used to code a range parameter.
- The 0-relative position of the 16 bytes within the table is indicated by the numeric suffix of the range parameter. For example, “**X30**” will define bytes 31 - 47 of a table.
- Any range parameter left out will default to the values in the base table.

Generating a new Case Translation Table

To assemble and link a new Case Translation Table, tailor the **ASMXLTB** member in the **CNTL** dataset. Run the job to create the new Case Translation Table.

The OLR Anchor Table (DBXANCHR)

The OLR Anchor Table (**DBXANCHR**) contains a list of all OLR system tables in use. This includes the System Parameters Table, the User Attribute Table, the Lock Table, the reentrant **DBXUTIL** module, and all OLR Language tables.

The Anchor Table ensures synchronization of access to all OLR system tables, and supports dynamic table replacement without the requirement for CICS system recycling .

You can choose to make many of the OLR system tables resident or non-resident in CICS. Through the OLR Anchor Table, you can specify if you want OLR to load a non-resident table , or to load a resident table once and thereafter access it by address.

By default, all OLR system tables are defined as resident to CICS, and each table will be loaded only once.

A copy of **DBXANCHR** is provided in the **TABLES** dataset in the installation/maintenance tape. It contains definitions for all system tables needed by the base system.

You may alter this copy and assemble it to generate a new copy of the Anchor Table. You **must not** delete any entries from the base table, and you **must not** change the entries for **DBXORUAT**, **DBXORLKT**, or **DBXUTIL**.

For each new language table to be used at your site, add a **TYPE=ANCHOR** entry to the OLR Anchor Table.

TYPE=ANCHOR parameters

Parameter	Description
ANCHOR	Specifies the OLR system table name. For your language table name, it can have up to eight characters (i.e. OLR Language Table name plus suffix).
LOAD	Specifies the initial load status for the table. There are two possibilities: REUSE This means that the table can be “reused” without reloading. LOAD This means that the table cannot be “reused” and must always be loaded prior to access.

Re-generating the OLR Anchor Table

To re-assemble and re-link the OLR Anchor Table, tailor the **ASMANCHR** member in the **CNTL** dataset. Run the job to update the OLR Anchor Table. The CICS region will need to be recycled to make the new **DBXANCHR** accessible.

The list of tables in the OLR Anchor Table displays with corresponding addresses and load statuses on the System Administration screen.

Use this screen to load a new copy of an updated table into CICS.

Installing a Language

Once you have defined a language in the System Parameters table, and modified the Language tables, you're ready to install the new language.

1. Make sure you have assembled and linked each table you have changed.
2. Re-assemble and re-link the System Parameters Table using a tailored version of **ASMORSIT** from the **CNTL** library.
3. If you have added any new tables, re-assemble and re-link the OLR System Anchor table.

To install a new OLR System Anchor table, you need to re-cycle the CICS region.

4. If you have not added any new tables, you can install new versions of language tables from the OLR System Administrator (OLRA) transaction:

From the OLRA menu screen, select System Administration. The OLR System Settings screen displays.

Press [F5] Newcopy to newcopy the tables and make them resident.

You cannot Newcopy resident tables with the CICS newcopy command. Instead, use the Newcopy function from the OLR System Settings screen.

Chapter 9 Working with External Topics

This chapter provides information about the External Topic Exit. With the External Topic Exit, you can attach programs which create topics from information you retrieve from an external file at the time the user asks to read the topic or note.

The External Topic Exit is useful when your application needs up to the minute information that is available from an external source. This chapter includes:

- Overview of the External Topic Exit
- Using the External Topic Exit
- Writing a External Topic Program

Overview

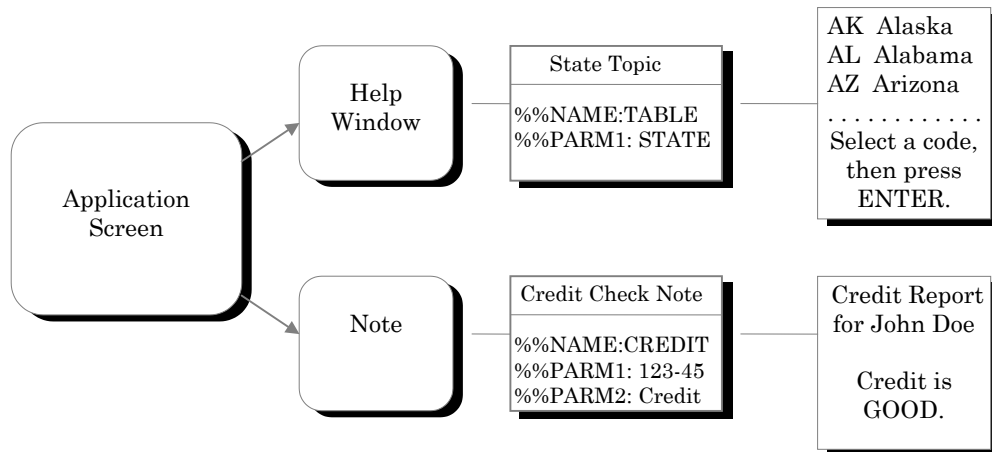
Some applications require information that frequently changes. If that information can be accessed from an external file, you can use the OLR External Topic Exit to retrieve and display it in an OnLine Reference topic, an OnLine Help window, or an OnLine Notepad note.

Instead of displaying the information stored in a topic in the OLR Server, an External Topic gets information from a file or database.

Documentation writers create external topics by typing a series of control statements into a topic.

These control statements are passed to a program which reads the external file and passes information to the OLR System.

The OLR System displays the information in a help topic, a reference topic, or a note.



External topics are useful when you need to display information that frequently changes, or information that is already available in another file or document management system.

Specifying an External Topic

Topics that use External Topic programs to provide their content include control statements instead of text. These control statements provide the name of the External Topic program, pass up to five parameters to the program, and specify handling of unsuccessful data requests.

When the OLR System receives a request for a topic which has these control statements, it runs the specified program and displays the results of that program as a topic or note.

Each control statement begins with a two character prefix. The standard prefix is '%%' unless it has been changed at your site.

When you use control statements, they must be on the first page of a topic. The control statements are:

Keyword	Required?	Value
NAME :	Yes	The name of the External Topic Program to be called to retrieve text for the topic (up to 8 characters). This is required and must be the first control statement.
ABEXIT :	No	Indicates the type of abend handling to be in effect. Valid values are: ON External Topic Program abends will be handled by the OLR System. If an abend is detected, the text retrieval process will be ended with a “failure” message. <i>ON is the default value.</i> OFF External Topic Program abends will not be handled. External Topic Program abends will cause the entire OLR session to end. This value should only be used when testing to force an abend dump.
ABEND :	No	Use of this keyword will force the OLR System to abend with the specified abend code on return from the External Topic Program. This is also for use in testing.
PARM1 :	No	The value associated with this keyword (up to 32 characters) is passed to the External Topic Program in the user parameter area of the Commarea. This value is passed by the OLR System to the External Topic Program
PARM2 :	No	Same as “PARM1”
PARM3 :	No	Same as “PARM1”
PARM4 :	No	Same as “PARM1”
PARM5 :	No	Same as “PARM1”

For example, the control statements:

```
%%NAME:      GETTABLE
%%PARM1:    STATES
```

run a program called 'GETTABLE' and passes it a PARM1 value of STATES. GETTABLE can be a general purpose program which accesses a number of tables. This control statement will cause it to branch to the logic it uses to access and format the STATES table.

Writing an External Topic Program

External Topic programs can be written in COBOL or Assembler, or any other language that can generate a CICS Command level program.

You can write separate programs for each external topic, or you can write programs that generate multiple topics. To write a program that generates multiple topics, use the PARM fields to pass information to the program to direct its processing.

Using the Sample Programs

Two sample programs are included in the **SAMPLES** dataset on the product installation tape:

- **DBXDP900** is a sample COBOL II program which reads an imaginary “state code” VSAM file and returns the records for a given state code. This program returns text lines using the system buffer provided in the commarea.
- **DBXDP920** is a sample Assembler program which reads an imaginary “state code” VSAM file and returns the records for a given state code. This program returns text lines in a user buffer provided by the External Topic Program.

Copy and rename one of the sample programs. If you are generating multiple topics in a single program, write the program logic needed to branch to a routine based on the value of the PARM which identifies the routine.

Decide whether you will pass the topic to OLR in the COMMAREA or in an external buffer. If you choose the external buffer, write the code to allocate and manage the buffer.

Write the data access routines for each table.

Review the Notes below and take the steps needed to add your program to the CICS and DB2 definitions.

Review the description of the fields in the COMMAREA (later in this section) to learn more about the testing options available via COMMAREA settings.

Notes about the External Topic program

1. The amount of text that the OLR System will accept from a External Topic Program is limited by the **ULPLIM** value specified, during installation, in the OLR System Parameters. When the number of lines returned exceeds the system limit, only the maximum allowable lines will be loaded into OLR dynamic storage and a “warning line” will be appended to the text.
2. In order to read an external file, it must be defined to the region where the OLR System is running.
3. If an External Topic Exit program retrieves information from a DB2 Table, the associated plan name must be connected to the transactions OLRC and ORXX in the RCT.
4. Two OLR System Parameters affect External Topic Exit processing:

The **ULPFIX** setting specifies the two-character prefix used to identify control statements

The **ULPLIM** setting specifies the maximum number of lines of text for a topic that can be retrieved by a External Topic Program. The default for this limit is 256. Setting the limit to zero has the effect of disabling the External Topic Exit.

Running an External Topic Program

External Topic Programs are a command-level CICS programs. When the user reads a topic which specifies an External Topic program, OLR uses a CICS link to invoke the program.

Input parameters are passed to the program through the Commarea. The External Topic program should have a copy of the Commarea in its Linkage section to provide addressability. Copylib members are included in the COPIES dataset on the OLR System Installation cartridge.

- UXP001 is a COBOL description of the Commarea
- UXP001B is an Assembler language description of the Commarea.

The External Topic Program Commarea

The Commarea includes four areas:

- the Input Parameter Area passes information between the OLR System and the External Topic program,
- the User Parameter Area passes the External Topic program control statements,
- the Output Parameter Area return information to OLR from the External Topic program,
- the System Text Buffer returns up to 48 lines of text from the External Topic program.

Input Parameter Area

Exit Name	Name of the External Topic Program
Return Code	<p>Normal Return - no errors so far, more text to return. The External Topic Program will be called again.</p> <p>End of Data - no errors, all text has been retrieved.</p> <p>No Data - no text was found.</p> <p>Failure - an error was encountered and the text could not be retrieved.</p>
Return Reason	Use together with “failure” return codes to indicate the reason for the invalid return. The four character “return reason” will be displayed in the error message.
Function	<p>Initial the first call to the External Topic Program.</p> <p>Continue subsequent calls to the External Topic Program</p>
Load Information	<p>Load Limit - maximum number of lines that will be accepted from a External Topic Program, as defined in the OLR System Parameters.</p> <p>Current Load - count of lines loaded so far.</p> <p>Maximum Return - maximum number of lines that may be returned in the commarea System Text Buffer. This maximum will not apply if an External Topic Buffer is used. Current maximum is 48.</p>
Intercept Information	<p>When the program is called by OnLine Help or OnLine Notepad, these fields contain information about the intercepted transaction:</p> <p>Transaction ID Screen ID Application ID Group ID PF Key invoking help or notepad.</p> <p>Cursor Row/Column Field Row/Column Field Length</p> <p>Field Value - content of field on application screen, up to 50 bytes. Field Qualifier - not currently used. Always spaces.</p>
Maximum User Parameters	This field is set to the maximum number of user parameters that can be carried in the User Parameter area. Current maximum is 5.

User Parameter Area

PARM1 Value	A 32 characters field containing the value passed as PARM1 If no PARM1 was passed, this field will contain spaces.
PARM2 Value	A 32 characters field containing the value passed as PARM2 If no PARM1 was passed, this field will contain spaces.
PARM3 Value	A 32 characters field containing the value passed as PARM3 If no PARM1 was passed, this field will contain spaces.
PARM4 Value	A 32 characters field containing the value passed as PARM4 If no PARM1 was passed, this field will contain spaces.
PARM4 Value	A 32 characters field containing the value passed as PARM5 If no PARM1 was passed, this field will contain spaces.

Output Parameter Area

Text Buffer Type	System Buffer - text lines are returned in the text buffer in the commarea. User Buffer - text lines are returned in a buffer at the address provided in the User Buffer Information field.
Text Page Size	The page size to be used to display the returned text lines. If zero, the page size of the topic will be used.
Text Line Size	The line size to be used to display the returned text lines. If zero, the line size of the topic will be used.
User Buffer Information	If text is being passed in a User Buffer: Buffer Address - the address of the External Topic Buffer.. Buffer Line Prefix Length - if using a control prefix on each text line, this value should be 3. Otherwise, it should be zero.
Return Count	The number of text lines returned in the System or User Buffer. If using the System Buffer, up to 48 text lines can be returned on each link to the External Topic Program.
Error Message	A 72 character error message, to be displayed in the message line on the terminal.
User Keys	A 64 byte area which can be used by the External Topic Program to hold control information over multiple links. This area is most useful as a place to save key information for the last row or record read in situations where more than one link to the External Topic Program is required to retrieve all of the text for a topic.

System Text Buffer

Text Line Prefix	<p>The following indicators are provided to set characteristics for the line:</p> <p>Display Break Indicator - causes the line on which it appears to be treated as 'top of page' when displayed.</p> <p>A value of 'Y' will cause the associated line to appear as the first line in any window or text screen in which it is displayed.</p> <p>Button Indicator - used to cause a 'button' to be placed immediately to the left of the indicated text line when displayed. The button makes it so that the cursor will move to the indicated line when the user presses the [TAB] key.</p> <p>A value of 'T' will cause the current line and all successive lines to be 'buttonized', until the button setting is turned off. The setting is turned off automatically when a blank line of text is reached, or when another button indicator set to 'P' is reached.</p> <p>Bright Indicator - causes the line on which it appears to be highlighted when displayed.</p> <p>A value of 'Y' will cause the associated line to appear as bright in any window or text screen in which it is displayed.</p>
Text Line	<p>Text, up to 72 characters in length. The length of the text line returned should correspond to the Line Size, either as specified by the External Topic Program or determined by default</p> <p>Text lines will not be reformatted by the OLR System. If the length of a text line exceeds the Line Size, truncation will occur.</p>

Using an External Topic Buffer

A External Topic program can return text lines in a buffer that it acquires. When the number of text lines would require multiple links to pass the data in the System Text Buffer, you can use the external buffer to perform a one-time Table load, then pass the result to OLR. OLR will manage the paging of data to the user screen.

The External Topic Program acquires the buffer using a GETMAIN, then passes the address of the buffer and a line count in the Output Parameter Area.

External Topic Buffer Format

The buffer should be prefixed by a fullword data area in which the first halfword is set to reflect the total buffer size. The first line of text starts just after the buffer prefix.

The External Topic Buffer can use of a text line prefix to control selective highlighting or "buttonization" of lines.

Unlike the System Text Buffer, which has fixed-length lines of 72 characters, the actual length of lines in the External Topic Buffer should be the same as the line size of the text. For example, if line size for the text is 32, then each line in the buffer should be 32 bytes in length.

Total buffer size can be calculated using the following formula:

$$4 + ((\text{Line prefix length} + \text{Line size}) * \text{number of lines})$$

For example, if line size for the text is 32 characters and 20 lines of text are returned, the total buffer size will be 644. If text line prefixes are used, an additional 3 bytes per line must be included. This would result in a total buffer size of 704.

External Topic program parameters

UXP001 COBOL Data Definition

```

05 UXP001-PARMS.
  10 UXP001-PARM-LENGTH      PIC S9(4)  VALUE +4144 COMP.
  10 FILLER                   PIC X(2)    VALUE SPACES.
  10 UXP001-EXIT-NAME        PIC X(8)    VALUE SPACES.
  10 UXP001-RTN-CODE         PIC S9(4)  VALUE +0    COMP.
    88 UXP001-NORM-RTN       VALUE +0.
    88 UXP001-END-DATA       VALUE +4.
    88 UXP001-NO-PROB       VALUE +0 +4.
    88 UXP001-NO-DATA       VALUE +8.
    88 UXP001-FAILURE       VALUE +12.
  10 UXP001-RTN-REASON       PIC X(4)    VALUE SPACES.
  10 FILLER                   PIC X(2)    VALUE LOW-VALUES.
    10 UXP001-INPUT-PARMS.
      15 UXP001-FUNCTION      PIC S9(4)  VALUE +0    COMP.
        88 UXP001-INITIAL-LOAD VALUE +0.
        88 UXP001-CONTINUE-LOAD VALUE +4.
      15 UXP001-TRANID       PIC X(4)    VALUE SPACES.
      15 UXP001-MAPNAME      PIC X(8)    VALUE SPACES.
      15 UXP001-APPLID       PIC X(8)    VALUE SPACES.
      15 UXP001-USERID       PIC X(8)    VALUE SPACES.
      15 UXP001-SYSID       PIC X(4)    VALUE SPACES.
      15 UXP001-AID         PIC X       VALUE SPACES.
      15 FILLER              PIC X       VALUE SPACES.
      15 UXP001-GROUP        PIC X(10)   VALUE SPACES.
      15 UXP001-LOAD-LIMIT   PIC S9(4)  VALUE +0    COMP.
      15 UXP001-LOAD-CURRENT PIC S9(4)  VALUE +0    COMP.
      15 UXP001-MAX-RTRN-CNT PIC S9(4)  VALUE +0    COMP.
        88 UXP001-MAX-RTN    VALUE +48.
      15 UXP001-CURSOR-ROW   PIC S9(4)  VALUE +0    COMP.
      15 UXP001-CURSOR-COL   PIC S9(4)  VALUE +0    COMP.
      15 UXP001-FLD-ROW     PIC S9(4)  VALUE +0    COMP.
      15 UXP001-FLD-COL     PIC S9(4)  VALUE +0    COMP.
      15 UXP001-FLD-LENGTH  PIC S9(4)  VALUE +0    COMP.
      15 UXP001-FLD-VALUE   PIC X(50)   VALUE SPACES.
      15 UXP001-FLD-QUAL    PIC X(10)   VALUE SPACES.
      15 UXP001-FLD-VALUE-EXT PIC X(50)  VALUE SPACES.
      15 UXP001-MAX-PARMS    PIC S9(4)  VALUE +0    COMP.
        88 UXP001-MAXP-VALUE VALUE +5.
      15 UXP001-INT-SAVE     PIC X(2)    VALUE LOW-VALUES.
      15 FILLER              PIC X(8)    VALUE LOW-VALUES.
  10 UXP001-USER-PARMS.
    15 UXP001-PARM-AREA      PIC X(160)  VALUE LOW-VALUES.
    15 UXP001-PARM-TAB
      REDEFINES UXP001-PARM-AREA OCCURS 5 TIMES
      INDEXED BY
      UXP001-PM-INDX.
    20 UXP001-USER-PRM PIC X(32).

```

UXP001 (continued)

```

10 UXP001-RETURN-PARMS.
   15 UXP001-RTN-TYPE      PIC S9(4)  VALUE +0    COMP..
      88 UXP001-SYSTEM-BUFFER      VALUE +0.
      88 UXP001-USER-BUFFER        VALUE +4.
   15 UXP001-PG-SIZE      PIC S9(4)  VALUE +0    COMP.
   15 UXP001-LN-SIZE      PIC S9(4)  VALUE +0    COMP.
   15 FILLER                PIC X(2)  VALUE LOW-VALUES.
   15 UXP001-USER-BUFFER-INFO.
      20 UXP001-UB-ADDR  PIC S9(9)  VALUE +0    COMP.
      20 UXP001-UB-PTR
          REDEFINES UXP001-UB-ADDR          POINTER.
      20 UXP001-UB-PLEN  PIC S9(4)  VALUE +0    COMP.
          88 UXP001-UB-NO-PREFIX      VALUE +0.
          88 UXP001-UB-PREFIX        VALUE +3.
          88 UXP001-UB-PREFIX-OK     VALUE +0 +3.
   15 UXP001-RETURN-COUNT PIC S9(4)  VALUE +0    COMP.
   15 UXP001-MESSAGE      PIC X(72)  VALUE SPACES.
   15 UXP001-USER-KEYS    PIC X(64)  VALUE SPACES.
   15 FILLER                PIC X(28)  VALUE LOW-VALUES.

10 UXP001-RETURN-LINES.
   15 UXP001-RETURN-LINE
                                     OCCURS 48 TIMES
                                     INDEXED BY
                                     UXP001-LN-INDX.

      20 UXP001-PREFIX-INFO.
          25 UXP001-TEXT-DBRK-IND
              PIC X.
              88 UXP001-TEXT-NO-DBRK  VALUE 'N'.
              88 UXP001-TEXT-DBRK     VALUE 'Y'.
          25 UXP001-TEXT-TABLE-IND
              PIC X.
              88 UXP001-TEXT-NO-TAB   VALUE 'N'.
              88 UXP001-TEXT-TABLE    VALUE 'T'.
              88 UXP001-TEXT-PARA     VALUE 'P'.
          25 UXP001-TEXT-BRT-IND
              PIC X.
              88 UXP001-TEXT-BRT-NO   VALUE 'N'.
              88 UXP001-TEXT-BRT-YES  VALUE 'Y'.
      20 UXP001-TEXT-LINE
          PIC X(72).

```

UXP001B Assembler DSECT

U001PARM	DS	0F	PARAMETER LIST BEGINNING
U001LL	DS	H	LIST LENGTH
	DS	H	RESERVED
U001EXIT	DS	CL8	EXIT NAME
	SPACE		
U001RTCD	DS	H	RETURN CODE, AS PER:
U001NORM	EQU	0	- NORMAL RETURN
U001EOD	EQU	4	- END DATA
U001NONE	EQU	8	- NO DATA FOUND
U001FAIL	EQU	12	- FAILURE
	SPACE		
U001REAS	DS	CL4	RETURN REASON
	SPACE		
	DS	CL2	RESERVED
	SPACE		
U001IPRM	DS	0F	INPUT PARAMETERS
	SPACE		
U001FUNC	DS	H	FUNCTION, AS PER:
U001INIT	EQU	0	- INITIAL LOAD = H'0'
U001CONT	EQU	4	- CONTINUE LOAD = H'4'
	SPACE		
U001TRAN	DS	CL4	TRANID INTERCEPTED
U001MAP	DS	CL8	INTERCEPTED MAPNAME
U001APPL	DS	CL8	INTERCEPTED APPLID
U001USER	DS	CL8	USERID
U001SYS	DS	CL4	SYSID OWNING INTERCEPTED TRANID
U001AID	DS	C	AID CAUSING INTERCEPTION
	DS	X	RESERVED
U001GRP	DS	CL10	GROUP
	SPACE		
U001LLIM	DS	H	LOAD LIMIT
U001LCUR	DS	H	CURRENT LINES LOADED
U001RMAX	DS	H	MAX LINES RETURNED - SYSTEM BUFFER
	SPACE		
U001CROW	DS	H	CURSOR ROW
U001CCOL	DS	H	CURSOR COLUMN
	SPACE		
U001FROW	DS	H	FIELD ROW
U001FCOL	DS	H	FIELD COLUMN
U001FLEN	DS	H	FIELD VALUE LENGTH
U001FVAL	DS	CL50	FIELD VALUE
U001FQ	DS	CL10	FIELD QUALIFIER
	SPACE		
U001FVAL	DS	CL50	FIELD VALUE - CURRENT
	SPACE		
U001PMAX	DS	H	MAXIMUM USER PARMS, AS PER:
U001PMUP	EQU	5	- ONLY FIVE ALLOWED

(continues. . .)

UXP001B (continued)

```

    SPACE
U001ISV DS XL2 INTERNAL SAVE AREA - NO MODIFY
        DS XL8 RESERVED
    SPACE
U001IPDL EQU *-U001PARM INPUT PARM TOTAL LENGTH
        SPACE 3
        DS 0F
U001UPS DS CL160 USER PARAMETERS
    SPACE
U001UPL EQU 32 USER PARM LENGTH
    SPACE
U001UPDL EQU *-U001UPS USER PARMS TOTAL LENGTH
        SPACE 3
U001RPRM DS 0F RETURNED PARAMETERS
    SPACE
U001RTYP DS H RETURN DATA FORMAT, AS PER:
U001SYSB EQU 0 - 'SYSTEM' BUFFERING
U001USRB EQU 4 - 'USER' BUFFERING
    SPACE
U001PGSZ DS H RETURN DATA PAGE SIZE, 1 - 15
U001LNSZ DS H RETURN DATA LINE SIZE, 1 - 72
    SPACE
        DS CL2 RESERVED
    SPACE
U001UBPT DS A -> USER BUFFER
U001UBPL DS H USER BUFFER LINE PREFIX, AS PER:
U001UBNP EQU 0 - NO PREFIX IN LINE
U001UBP EQU 3 - 3-BYTE PREFIX ON EACH LINE
    SPACE
U001RET# DS H LINES OF TEXT RETURNED THIS PASS
    SPACE
U001MSG DS CL72 MESSAGE TEXT RETURNED
U001UKEY DS CL64 USER KEYS
        DS CL28 RESERVED
U001RPDL EQU *-U001RPRM RETURNED PARMS MAX LENGTH
        SPACE 3
U001RBLK DS CL3600 RETURNED TEXT LINES
    SPACE
U001RBDL EQU *-U001RBLK RETURNED TEXT MAX LENGTH
U001PTDL EQU *-U001PARM TOTAL PARM LENGTH S/B 4144
        SPACE 3
*-----*
* U001TLDS - DESCRIBE ONE TEXT LINE SLOT IN BLOCK *
*-----*
        SPACE 3
U001TLDS DSECT RETURNED TEXT LINE DESCRIPTION
U001TPFX DS 0C LINE PREFIX, AS PER:
        SPACE

```

(continues. . .)

UXP001B (continued)

```

U001DBRK DS C - DISPLAY BREAK INDICATOR, AS PER:
U001DBY EQU C'Y' . THIS LINE IS A DBRK
U001DBN EQU C'N' . THIS LINE IS NOT A DBRK
    SPACE
U001TAB DS C - TABLE INDICATOR, AS PER:
U001TBN EQU C'N' . THIS LINE NOT TABLE TYPE
U001TBY EQU C'T' . THIS LINE IS TABLE TYPE

```

```
U001PGY EQU C'P'           . THIS LINE IS NEW PARAGRAPH
        SPACE
U001BRTI DS C              - LINE HILIGHT INDICATOR, AS PER:
U001BRTY EQU C'Y'         . THIS LINE IS ALL BRIGHT
U001BRTN EQU C'N'         . THIS LINE IS NORMAL
        SPACE
U001TXT DS CL72           TEXT GOES HERE, UP TO 72 BYTES
        SPACE
U001TLDL EQU *-U001TLDS  LENGTH OF ONE LINE, PREFIX + DATA
```


Chapter 10 OLR System Exits

This chapter provides information about the exits available to allow you to tailor the OLR System to fit your site's application needs:

- the Intercept Exit
- the OLR Exit
- the User Profile Exit
- the Note Form Selection Exit
- the Note Field Validation Exit
- the Terminal Buffer Text Stream Extract Utility

Samples of all of these exits is provided in the **SAMPLES** library on the product distribution cartridge. The copy members for the exit parameter lists are in the **COPIES** library.

The Intercept Exit (DBXUXINT)

The Intercept Exit gives you control as soon as the OLR Intercept determines that an OnLine Help or OnLine Notepad service has been requested by a CICS transaction.

At this exit, you can examine information about the active transaction, the user, and the information the OLR System is preparing to display.

You can change the information, then return control to the OLR System. This is useful when you want to make a programmatic decision about information to be displayed.

- you can change the Application specified in the help link to provide tailored information based on information gathered from or derived from the transaction screen.
- you can change the Notepad topic from the value at the cursor location to a value you gather or derive from information on the transaction screen.
- you can cause a Notepad user to be prompted with an OLR Form, rather than a blank note entry screen. You can also pre-prime the data entry fields in the form based on buffer contents, etc.

you can also use this exit to perform additional security checking, to gather statistics, or to terminate the request.

Using the Intercept User Exit

To use the Intercept Exit, replace the DBXUXINT module linked into the OLR System Entry Point Program with a command-level CICS program written at your site, then set the **UXINT** parameter in the System Parameters Table to "YES".

Your DBXUXINT program may be written in Assembler or Cobol II.

Some programming considerations:

1. Though your DBXUXINT must be a command-level program, it will be invoked by a static call, not a CICS link. This means that, while you will be able to use CICS commands within your program, you should not use a CICS RETURN to return control to the OLR System.
2. For DBXUXINT programs written in COBOL II, use GOBACK to return to the OLR System. For DBXUXINT programs written in assembler, use a standard OS return.

See the ASMUXINT and COBUXINT members in the **SAMPLES** dataset from the installation tape for examples of skeleton DBXUXINT programs in Assembler and Cobol II.

An optional utility, DBXEXTDS, has been provided to enable you to extract the content of any field in the intercepted screen. This program is described later in this chapter.

Exit Parameters

The OLR System communicates with the your DBXUXINT program through parameters passed in the commarea.

The member **UXP002** in the **COPIES** dataset from the installation tape contains a commarea copy definition for a DBXUXINT program written in COBOL. Use UXP002B for a DBXUXINT program written in Assembler.

Linking your program

To link your DBXUXINT into the OLR System, re-link program DBXPR000 using the control cards provided in the LINKCNTL library on your installation cartridge. Make sure that the object module for your version of DBXUXINT is the one that will be picked up in the link.

DBXUXINT Parameters

Return Code	<p>Returned to the OLR System by the DBXUXINT program. Valid values are:</p> <p>Normal Return - no errors.</p> <p>Access Denied - user will be returned to application from which help or note key was pressed. No message.</p> <p>Failure - the DBXUXINT program has detected an error condition that prevents it from completing successfully. The user will see an error message, then be returned to the application from which help or note key was pressed.</p> <p>Failure with dump- same as above. A dump will be generated.</p>
Return Reason	<p>Use together with “failure” or “failure with dump” return codes to indicate the reason for the invalid return. The four character “return reason” will be displayed in the error message.</p>
Input Parameters	<p>These fields contain information about the intercepted transaction:</p> <p>Intercept Function - Help or Notepad</p> <p>Transaction ID</p> <p>Cursor Row/Column - position of cursor when help or notepad PF Key was pressed.</p> <p>Field Row/Column - position of field on application screen.</p> <p>Field Length - length of field on application screen.</p> <p>PF Key invoking help or notepad.</p>
Forms Parameters	<p>Note TTL - specifies the note title to be used</p> <p>Note Qual - specifies the note qualifier to be used</p> <p>Form Name - specifies the name of the form to be used</p> <p>Form Qual - specifies the qualifier of the note form name</p> <p>FFV Count - number of field values being passed to OnLine Notepad</p> <p>FFV QID - the Queue ID of the field value queue</p>

Return Parameters	<p>These fields contain information about the intercepted transaction that can be overridden by the DBXUXINT program:</p> <p>User ID</p> <p>Application ID</p> <p>Screen ID</p> <p>Group</p> <p>PF Key invoking help or notepad.</p> <p>Field Value - content of field where the cursor was on application screen.</p> <p>Field Qualifier - not currently used. Always spaces.</p> <p>Notepad Topic Protect indicator - indicates whether the title and qualifier fields in the OnLine Notepad window will be protected from being changed. This will also suppress the Topics List function.</p>
--------------------------	---

The OLR System Exit (DBXUXOLR)

The OLR System Exit gives you control whenever a user requests entry into the OLR System using the System Administration transaction (OLRA), the OnLine Reference transaction (OLRX), or the OnLine Help Link List transaction (OLHX).

At this exit, you can override the active CICS Userid. Since the CICS Userid is used to grant access to information in the OLR System, you can dynamically control access by changing the Userid.

Using the OLR System Exit

To use the OLR System Exit, replace the DBXUXOLR module linked into the OLR System Entry Point Program with a command-level CICS program written at your site. You will also need to set the **UXOLR** parameter in the System Parameters Table to “YES”.

Your DBXUXOLR program may be written in Assembler or Cobol II.

Some programming considerations:

1. Though your DBXUXOLR must be a command-level program, it will be invoked by a static call, not a CICS link. This means that, while you will be able to use CICS commands within your program, you should not use a CICS RETURN to return control to the OLR System.
2. For DBXUXOLR programs written in COBOL II, use GOBACK to return to the OLR System. For DBXUXOLR programs written in assembler, use a standard OS return.

See the ASMUXOLR and COBUXOLR members in the CARDS dataset from the installation tape for examples of skeleton DBXUXOLR programs in Assembler and Cobol II.

Exit Parameters

The OLR System communicates with your DBXUXOLR program through parameters passed in the commarea.

The member **UXP005** in the **COPIES** dataset from the installation tape contains a commarea copy definition for a DBXUXOLR program written in COBOL. Use **UXP005B** for a DBXUXOLR program written in Assembler.

Linking your program

To link your DBXUXOLR into the OLR System, re-link program DBXPR000 using the control cards provided in the **LINKCNTL** library on the installation tape. Make sure that the object module for your version of DBXUXOLR is the one that will be picked up in the link.

DBXUXOLR Parameters

Return Code	<p>Returned to the OLR System by the DBXUXINT program. Valid values are:</p> <p>Normal Return - no errors.</p> <p>Access Denied - user will be returned to application from which help or note key was pressed. No message.</p> <p>Failure - the DBXUXOLR program has detected an error condition that prevents it from completing successfully. The user will see an error message, and the OLR session will be terminated</p> <p>Failure with dump- same as above. A dump will be generated.</p>
Return Reason	<p>Use together with “failure” or “failure with dump” return codes to indicate the reason for the invalid return. The four character “return reason” will be displayed in the error message.</p>
Input Parameters	<p>These fields contain information about the intercepted transaction:</p> <p>Initial entry tranid - OLRA, OLRX or OLHX transaction entry</p>
Return Parameters	<p>User ID</p>

The User Profile Exit (DBXUXUSR)

The User Profile exit gives you control at the point when the OLR System has checked user access levels. This exit provides information about user access to OnLine Help, OnLine Notepad, OnLine Reference and the OLR System Administrators transaction.

At this exit, you can change the settings of any user's preferences or system privileges. For example,

- you can assign a Language based on the transaction, application or screen the user is in.
- you can change the help mode based on the current userid, application, screen or PF key that was pressed.

Using the User Profile Exit

To use the User Profile Exit, replace the DBXUXUSR module linked into the OLR System Entry Point Program, OnLine Help Program and OnLine Notepad Program with a command-level CICS program written at your site. You will also need to set the **UXUSR** parameter in the System Parameters Table to "YES".

Your DBXUXUSR program may be written in Assembler or Cobol II.

Some programming considerations:

1. Though your DBXUXUSR must be a command-level program, it will be invoked by a static call, not a CICS link. This means that, while you will be able to use CICS commands within your program, you should not use a CICS RETURN to return control to the OLR System.
2. For DBXUXUSR programs written in COBOL II, use GOBACK to return to the OLR System. For DBXUXUSR programs written in assembler, use a standard OS return.

See the ASMUXUSR and COBUXUSR members in the CARDS dataset from the installation tape for examples of skeleton DBXUXUSR programs in Assembler and Cobol II.

An optional utility, DBXEXTDS, has been provided to enable you to extract the content of any field in the intercepted screen when accessing OnLine Help or OnLine Notepad. This program is described later in this chapter.

Exit Parameters

The OLR System communicates with the your DBXUXUSR program through parameters passed in the commarea.

The member **UXP004** in the **COPIES** dataset from the installation tape contains a commarea copy definition for a DBXUXUSR program written in COBOL. Use **UXP004B** for a DBXUXUSR program written in Assembler.

Linking your program

To link your DBXUXUSR into the OLR System , re-link programs DBXPR000, DBXPR030 and DBXWH360 using the control cards provided in the **LINKCNTL** library from the installation tape. Make sure that the object module for your version of DBXUXUSR is the one that will be picked up in the link.

DBXUXUSR Parameters

Return Code	<p>Returned to the OLR System by the DBXUXINT program. Valid values are:</p> <p>Normal Return - no errors.</p> <p>Changes made - changes have been made in the parameters that will be returned from DBXUXUSR. No errors.</p> <p>Failure - the DBXUXUSR program has detected an error condition that prevents it from completing successfully. The user will see an error message, and the OnLine Help, OnLine Notepad or the OLR session will be terminated</p>
Return Reason	<p>Use together with “failure” return codes to indicate the reason for the invalid return. The four character “return reason” will be displayed in the error message.</p>
Input Parameters	<p>These fields contain general and intercept-related information about the transaction invoked:</p> <p>Function - stand-alone (direct), attach, Help or Notepad entry</p> <p>Userid</p> <p>Transaction ID</p> <p>Cursor Row/Column - position of cursor when PF Key was pressed.</p> <p>Field Row/Column - position of field on application screen.</p> <p>Field Length - length of field on application screen.</p> <p>PF Key invoking help or notepad.</p> <p>Application ID</p> <p>Screen ID</p> <p>Group</p> <p>PF Key invoking help or notepad.</p> <p>Field Value - content of field at the cursor.</p> <p>Field Qualifier - not currently used. Always spaces.</p>
Return Parameters	<p>These fields contain information about the current user’s registered user profile that can be overridden by the DBXUXUSR program:</p> <p>Default Group</p> <p>Title Search Privilege</p> <p>Author/Date Search Privilege</p> <p>Keyword Search Privilege</p> <p>Extended Find Privilege</p> <p>Display Qualifier Preference</p> <p>Paging Prompt Preference</p> <p>User Administrator Authority</p> <p>User Administrator Grant Authority</p> <p>System Administrator Authority</p> <p>System Administrator Grant Authority</p> <p>Language Key</p> <p>Help Mode</p>

The Note Form Selection Exit (DBXUXNFF):

The Note Form Selection Exit gives you control whenever an 'ADDNOTE' window is filled in by a user when beginning to write a note.

At this exit, you can examine information about the active transaction, the user, and the note the user is preparing to write.

You can change the information, then return control to the OLR System.

- you can change the Form the user has selected.
- you can cause a Form to be used, whether the user requested one or not.
- you can cause Form processing to be ignored totally.
- you can generate field values to pre-prime the Form when displayed to the user.

Using the Note Form Selection Exit

To use this Exit, replace the DBXUXNFF module linked into the OLR System Program DBXPR320 with a command-level CICS program written at your site.

Your DBXUXNFF program may be written in Assembler or Cobol II.

Some programming considerations:

1. Though your DBXUXNFF must be a command-level program, it will be invoked by a static call, not a CICS link. This means that, while you will be able to use CICS commands within your program, you should not use a CICS RETURN to return control to the OLR System.
2. For DBXUXNFF programs written in COBOL II, use GOBACK to return to the OLR System. For DBXUXNFF programs written in assembler, use a standard OS return.

See the DBXUXNFF member in the **SAMPLES** dataset from the installation tape for an example of a DBXUXNFF program.

Exit Parameters

The OLR System communicates with the your DBXUXNFF program through parameters passed in the commarea.

The member **UXP007** in the **COPIES** dataset from the installation cartridge contains a commarea copy definition for a DBXUXNFF program written in COBOL. Use **UXP002B** for an exit program written in Assembler.

Linking your program

To link your DBXUXNFF into the OLR System, re-link program DBXPR320 using the DBXPR320 link control cards in the LINKCNTL library of your installation cartridge. Make sure that the object module for your version of DBXUXNFF is the one that will be picked up in the link.

DBXUXNFF Parameters

Return Code	<p>Returned to the OLR System by the DBXUXNFF program. Valid values are:</p> <p>Normal Return - no errors, ignore form processing.</p> <p>Use Form - use Form provided below.</p> <p>Failure - the DBXUXNFF program has detected an error condition that prevents it from completing successfully. The user will see an error message.</p>
Return Reason	<p>Use together with "failure" return code to indicate the reason for the invalid return. The four character "return reason" will be displayed in the error message.</p>
Input Parameters	<p>These fields contain information about the current OLR transaction: (If entry to OLR was not via the intercept, intercept related fields below will be blank or null).</p> <p>Transaction ID</p> <p>Screenid</p> <p>Cursor Row/Column - position of cursor when help or notepad PF Key was pressed.</p> <p>Field Row/Column - position of field on application screen.</p> <p>Field Length - length of field on application screen.</p> <p>Intercept QID - the ID of the queue containing the intercepted application screen buffer.</p> <p>PF Key invoking help or notepad.</p>
Return Parameters	<p>Form Name - name of form to be used</p> <p>Form Qual - specifies the qualifier of the note form name</p> <p>FFV Count - number of field values being passed to OnLine Notepad</p> <p>FFV QID - the Queue ID of the field value queue</p>

Note Form Field Validation Exits

The definition of a Form Field includes the optional name of a validation exit that may be invoked to edit the contents of the field as entered by the user filling out the form.

These validation exits are CICS programs, and are invoked via EXEC CICS LINK.

Parameters are passed to a validation exit via the CICS commarea. The format of the commarea is described by copy member **UXP006** in the **COPIES** library unloaded from your OLR install cartridge.

Basically, the parameters passed to a validation exit are the 32-byte name of the field to edit, and the actual value (up to 50 bytes) entered by the user.

The exit program can edit the data in any way it chooses, and then return a pass or fail indicator. An optional 48 bytes explaining the reason for failure may also be returned, and this will be displayed on the terminal.

Three sample validation programs are included in the **SAMPLES** library on the OLR install cartridge. These are **DBXEDALP**, **DBXEDNUM**, and **DBXEDVFD**.

These first two validation exits are also distributed with the product as executables, and provide the default Alpha and Numeric edits for form fields. **DBXEDVFD** is a true sample, and is only included as source in **SAMPLES**.

Using the Form Field Validation Exit

Validation exits may be written in Assembler or Cobol II.

Programming considerations:

1. A Validation exit must be a CICS command-level program, invoked by EXEC CICS LINK.
2. The commarea passed to a validation exit is described by **UXP006** (**UXP006B** for assembler) from the **COPIES** library..
3. If you need to access DB2 from a validation exit, you need to include the exit name in the bind for OLR plan **XTSONL**.

Field Validation Exit Parameters

Return Code	Returned to the OLR System by the validation program. Valid values are: Normal Return - no errors, ignore form processing. Failure - the exit has detected an error condition that prevents it from completing successfully. The user will see an error message.
Return Reason	Use together with “failure” return code to indicate the reason for the invalid return. The 48-byte reason will be displayed in an error message.
Input Parameters	Field name - the 32 byte name of the field.
	Field data length - the length of data to edit.
	Field Value - the value to edit, 50 bytes length max.

Using DBXEXTDS - Data Stream Extract Utility

An optional utility, DBXEXTDS, has been provided to enable you to extract the content of any field in the intercepted screen when using OnLine Help or OnLine Notepad. Called from your Intercept Exit, User Profile Exit or External Topic Exit program, DBXEXTDS can be used:

- to derive a Notepad topic from fields other than the one the cursor was on, or
- to vary the help provided for a field depending on the contents of another field on the screen, or
- to retrieve text information based on a particular field value.

How to call DBXEXTDS from your Exit programs:

1. Your Intercept User Exit program (DBXUXINT), User Profile User Exit program (DBXUXUSR), Note Form Selection Exit program (DBXUXNFF), or your User Text Exit program communicates with the DBXEXTDS utility through a set of parameters.

The member **UXP003** in the **COPIES** dataset contains a parameter copy definition for a DBXUXINT program written in COBOL. Use **UXP003B** for a DBXUXINT program written in Assembler. See Appendix A of this guide for listings of these copy members.

2. Prior to calling DBXEXTDS it is the responsibility of the User Exit program to retrieve the temporary storage record containing the data stream to be queried. The address of the retrieved record is then passed to the DBXEXTDS utility in the UXP003-QUEUE-ADDR field. This can be accomplished using the following CICS command:

From the Intercept User Exit program DBXUXINT:

```
EXEC CICS READQ TS
      QUEUE      (UXP002-INT-QID)
      SET        (UXP003-QUEUE-PNTR)
      LENGTH     (WS-LENGTH)
END-EXEC.
```

From the User Profile User Exit program DBXUXUSR:

```
EXEC CICS READQ TS
      QUEUE      (UXP004-INT-QID)
      SET        (UXP003-QUEUE-PNTR)
      LENGTH     (WS-LENGTH)
END-EXEC.
```

From a User Text Exit program:

```
EXEC CICS READQ TS
      QUEUE      ('OLR1XXXX')
      SET        (UXP003-QUEUE-PNTR)
      LENGTH     (WS-LENGTH)
END-EXEC.
```

where 'OLR1' is a constant value and 'XXXX' is the active terminal id.

Other information that must be passed to DBXEXTDS:

UXP003-EXT-ROW	- row number of field on the screen from which data is to be extracted
UXP003-EXT-COL	- starting column of field on the screen from which data is to be extracted

UXP003-LINE-SIZE - screen width of the application OLR is windowing over (generally 80).

DBXEXTDS is invoked using a standard call:

CALL 'DBXEXTDS' USING UXP003-PARMS.

On return, UXP003-RTN-VALUE will contain the contents of the requested field. DBXEXTDS may be called more than once to retrieve the contents of multiple fields.

Linking DBXEXTDS with your program

Include DBXEXTDS in the link control cards for your user exit program.

For the Intercept User Exit program, you will need to re-link program DBXPR000 to include the DBXEXTDS utility. Add DBXEXTDS to the link control cards for DBXPR000 provided in the **DBXLINK1** member of your installation tape or the most recent maintenance tape before relinking.

For the User Profile User Exit program, you will need to re-link programs DBXPR000, DBXPR030 and DBXWH360 to include the DBXEXTDS utility. Add DBXEXTDS to the link control cards for DBXPR000, DBXPR030 and DBXWH360 provided in the **DBXLINK1** member of your installation tape or the most recent maintenance tape before relinking.