Fifth Edition (November 2005)
This edition replaces the Version 6.1 edition.

© Copyright GXS, Inc. 1998, 2005. All rights reserved.
Government Users Restricted Rights - Use, duplication, or disclosure restricted.
# Contents

To the reader ................................................................. ix  
Who should read this book ...................................................... ix  
What this book covers .......................................................... ix  
How this book is organized ..................................................... x  
Type conventions ............................................................... x  
How information is presented ............................................... xi  
Related books .................................................................... xii 

Chapter 1. Getting started ...................................................... 1  
ExpOpenCInterface ............................................................... 2  
ExpCloseCInterface .............................................................. 3  
Error information ................................................................... 4  
ExpErrorExplanation ............................................................ 4  
Miscellaneous tools ............................................................... 6  
ExpAddField ........................................................................ 6  
ExpReplaceField ................................................................... 7  
ExpResetMem ......................................................................... 8  
ExpSizeOfOrdr ....................................................................... 9  
ExpSizeOfRept ...................................................................... 10  

Chapter 2. Writing the interface for the Projects database .......... 11  
Projects ............................................................................... 13  
ExpOpenProj ....................................................................... 13  
ExpCloseProj ....................................................................... 16
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpAddProj</td>
<td>17</td>
</tr>
<tr>
<td>ExpAsgProjToCommProf</td>
<td>24</td>
</tr>
<tr>
<td>ExpDeleteProj</td>
<td>26</td>
</tr>
<tr>
<td>ExpListProj</td>
<td>27</td>
</tr>
<tr>
<td>ExpListProjOnCommProf</td>
<td>32</td>
</tr>
<tr>
<td>ExpReplaceProj</td>
<td>37</td>
</tr>
<tr>
<td>ExpUsgProjFromCommProf</td>
<td>43</td>
</tr>
<tr>
<td>Communications profiles</td>
<td>45</td>
</tr>
<tr>
<td>ExpAddCommProf</td>
<td>45</td>
</tr>
<tr>
<td>ExpDeleteCommProf</td>
<td>50</td>
</tr>
<tr>
<td>ExpListCommProf</td>
<td>52</td>
</tr>
<tr>
<td>ExpReplaceCommProf</td>
<td>58</td>
</tr>
<tr>
<td>Logon IDs</td>
<td>63</td>
</tr>
<tr>
<td>ExpAddIeLogon</td>
<td>63</td>
</tr>
<tr>
<td>ExpDeleteIeLogon</td>
<td>68</td>
</tr>
<tr>
<td>ExpListIeLogon</td>
<td>70</td>
</tr>
<tr>
<td>ExpReplaceIeLogon</td>
<td>77</td>
</tr>
<tr>
<td>ExpRetrieveErrText</td>
<td>82</td>
</tr>
</tbody>
</table>

Chapter 3. Writing the interface for the AddressBook database

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpOpenAdbk</td>
<td>89</td>
</tr>
<tr>
<td>ExpCloseAdbk</td>
<td>90</td>
</tr>
<tr>
<td>Addresses</td>
<td>91</td>
</tr>
<tr>
<td>ExpAddAddr</td>
<td>91</td>
</tr>
<tr>
<td>ExpAsgAddrToDistList</td>
<td>105</td>
</tr>
<tr>
<td>ExpAsgAddrToTradProf</td>
<td>107</td>
</tr>
<tr>
<td>ExpDeleteAddr</td>
<td>109</td>
</tr>
<tr>
<td>ExpListAddr</td>
<td>111</td>
</tr>
<tr>
<td>ExpListAddrOnDistList</td>
<td>121</td>
</tr>
<tr>
<td>ExpListAddrOnTradProf</td>
<td>130</td>
</tr>
<tr>
<td>ExpReplaceAddr</td>
<td>139</td>
</tr>
<tr>
<td>ExpUsgAddrFromDistList</td>
<td>153</td>
</tr>
<tr>
<td>ExpUsgAddrFromTradProf</td>
<td>155</td>
</tr>
<tr>
<td>Distribution lists</td>
<td>157</td>
</tr>
<tr>
<td>ExpAddDistList</td>
<td>157</td>
</tr>
<tr>
<td>ExpAsgDistListToTradProf</td>
<td>160</td>
</tr>
<tr>
<td>ExpDeleteDistList</td>
<td>162</td>
</tr>
<tr>
<td>ExpListDistList</td>
<td>164</td>
</tr>
<tr>
<td>ExpListDistListsOnTradProf</td>
<td>169</td>
</tr>
<tr>
<td>ExpReplaceDistList</td>
<td>173</td>
</tr>
</tbody>
</table>
ExpUsdgDistListFromTradProf .................................................. 176
Trading profiles ................................................................. 178
ExpAddTradProf ................................................................. 178
ExpDeleteTradProf ............................................................. 189
ExpListTradProf ................................................................. 191
ExpReplaceTradProf ........................................................... 200

Chapter 4. Writing the interface for the MailBox database .................... 211
MailBox .................................................. 213
ExpOpenMbox .................................................. 213
ExpCloseMbox .................................................. 214
Order shelf and receipt shelf ............................................. 215
ExpListOrdrsOnOrdrShlf .................................................. 215
ExpListRcptsOnSessRcpt .................................................. 221
Dropoff boxes .................................................. 225
ExpAddDropBox .................................................. 225
ExpDeleteDropBox .................................................. 242
ExpListDropBox .................................................. 244
ExpReplaceDropBox .................................................. 257
Receive orders .................................................. 274
ExpAddRecvOrdr .................................................. 274
ExpAsgRecvOrdrToDropBox .................................................. 302
ExpDeleteRecvOrdr .................................................. 304
ExpListRecvOrdr .................................................. 306
ExpReplaceRecvOrdr .................................................. 322
ExpUsdgRecvOrdrFromDropBox .................................................. 348
Send orders .................................................. 350
ExpAddSendOrdr .................................................. 350
ExpAsgSendOrdrToDropBox .................................................. 378
ExpDeleteSendOrdr .................................................. 380
ExpListSendOrdr .................................................. 382
ExpReplaceSendOrdr .................................................. 399
ExpUsdgSendOrdrFromDropBox .................................................. 426
Session receipts .................................................. 428
ExpDeleteSessRept .................................................. 428
ExpListSessRept .................................................. 430
Receive receipts .................................................. 439
ExpListRecvRept .................................................. 439
Send receipts .................................................. 460
ExpListSendRept .................................................. 460
Chapter 5. Writing the interface for Information Exchange .................................. 473
Communicating with Information Exchange ..................................................... 474
ExpOpenMSys ............................................. 474
ExpCloseMSys ........................................... 475
Processing dropoff boxes in batch mode ......................................................... 476
ExpProcDropBox ........................................... 476
Starting and ending Information Exchange sessions .................................... 481
ExpEndSess .................................................. 481
ExpStartSess .............................................. 487
Recovering Information Exchange sessions ................................................... 498
ExpRecoverSessRcpt ....................................... 498
Interactive session commands ................................................................. 504
ExpDoPurgOrdr ............................................. 504
ExpDoQuryOrdr ............................................. 507
ExpDoRecvOrdr ............................................. 518
ExpDoSendOrdr ............................................. 536

Appendix A. The Java interface classes ......................................................... 553
Database classes, members, and methods .................................................... 553
public class ExpAddress .............................................. 555
public class ExpAddressList extends LinkedList ....................................... 557
public class ExpCommProfile ......................................... 558
public class ExpCommProfileList extends LinkedList ............................... 559
public class ExpDBLister extends LinkedList ......................................... 560
public class ExpDistributionList ........................................ 563
public class ExpDistListList extends LinkedList ..................................... 564
public class ExpDropBox ............................................. 565
public class ExpDropBoxList extends LinkedList ..................................... 567
public class ExpException extends Exception ....................................... 568
public class ExpIeLogon ............................................. 569
public class ExpIeLogonList extends LinkedList ..................................... 570
public class ExpJni ................................................. 571
public class ExpMailboxItem .......................................... 572
public class ExpMailboxItemList extends LinkedList ................................ 573
public class ExpOrderList extends LinkedList ....................................... 574
public class ExpProject ............................................. 575
public class ExpProjectList extends LinkedList ..................................... 576
public class ExpReceiveOrder ........................................ 577
public class ExpReceiveReceipt ........................................ 579
public class ExpReceiptList extends LinkedList ..................................... 581
public class ExpReceiveOrderList extends LinkedList ............................ 582
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>public class ExpReceiveReceiptList</td>
<td>583</td>
</tr>
<tr>
<td>public class ExpSendOrder</td>
<td>584</td>
</tr>
<tr>
<td>public class ExpSendReceipt</td>
<td>586</td>
</tr>
<tr>
<td>public class ExpSendOrderList</td>
<td>587</td>
</tr>
<tr>
<td>public class ExpSendReceiptList</td>
<td>588</td>
</tr>
<tr>
<td>public class ExpSessionReceipt</td>
<td>589</td>
</tr>
<tr>
<td>public class ExpSessionReceiptList</td>
<td>590</td>
</tr>
<tr>
<td>public class ExpSession</td>
<td>591</td>
</tr>
<tr>
<td>public class ExpTradingProfile</td>
<td>593</td>
</tr>
<tr>
<td>public class ExpTradingProfileList</td>
<td>594</td>
</tr>
</tbody>
</table>

**Glossary** ................................................................. 595

**Index** ................................................................. 603
To the reader

The term network is used in this book to refer to the worldwide communications network infrastructure provided by AT&T Global Network Services.

Who should read this book

This book is intended as a reference for programmers who want to use the Expedite for Windows™ C-language interface, or the Java interface.

What this book covers

Expedite for Windows is an Information Exchange communications program that enables you to transmit data files and messages to and from Information Exchange. This book gives you the information necessary to program and use Expedite for Windows for your company’s applications.
How this book is organized

The book is organized as follows:

- Chapter 1, “Getting started,” provides the startup information needed to write your interface using Expedite for Windows.
- Chapter 2, “Writing the interface for the Projects database,” provides reference information for project functions you use in the Projects database.
- Chapter 3, “Writing the interface for the AddressBook database,” provides reference information for address functions you use in the AddressBook database.
- Chapter 4, “Writing the interface for the MailBox database,” provides reference information for mailbox functions you use in the MailBox database.
- Chapter 5, “Writing the interface for Information Exchange,” provides reference information for Information Exchange interactive functions.
- Appendix A, “The Java interface classes,” provides reference information for the Java interface database classes, members, and methods.

This book also contains a notices section, a glossary, and an index.

Glossary words are shown in italics the first time they are used in this book.

Type conventions

The following type conventions are used in this book:

- All functions are in bold, mixed case, blue type.
- All prototypes and arguments are in monospaced font.
- All field names are in bold, mixed case, black type.
- All field descriptions are in paragraph format.
- All field options are in table format.
How information is presented

Following are examples of how information is presented in this book.

ExpOpenProj

This function opens a project. If a Project is already open, the return code is set to 4. If the project name specified is not found, the return code is set to 8. Otherwise, the project is opened and saved for use on the OpenAddrBook, the OpenMailbox, and the OpenMsgSys functions for this project.

Function prototype

```c
int ExpOpenProj(
    struct ExpOpenProjReqStruct * Input,
    struct ExpProjOpenedRspStruct * Output
);
```

Input arguments

```c
struct ExpOpenProjReqStruct
{
    /* Project password */
    char ProjPswd[EXP_MAXLEN_PJ_PROJ_PSWD];
    /* Name of the project */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpOpenProjReqStruct

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must exist.
Related books

You may find the following publications helpful when performing the tasks described in this book:

- *Information Exchange Mailbox Command Reference*, GC34-2260
- *Information Exchange Messages and Formats*, GC34-2324
- *Information Exchange Message Charges Reference*, GX66-0653

These books are also available on the library page of the EDI Services Web site at [http://www.gxsolc.com/edi_bes.html](http://www.gxsolc.com/edi_bes.html).
Chapter 1

Getting started

Expedite for Windows is an interface to Information Exchange. This programming reference provides the information you need to write your interface using C-language functions (commands). In addition, this book provides details on the Java interface that uses the C-language functions. (See Appendix A, “The Java interface classes,” for more information.)

The information in this chapter lists the C-language functions needed to get started. For complete information about programming your C-language interface for Expedite for Windows, see the Expedite for Windows Software Development Kit Programming Guide.

The functions discussed in this chapter are:

ExpOpenCInterface ................................................................. 2
ExpCloseCInterface ................................................................. 3

Error information ................................................................. 4
  ExpErrorExplanation ......................................................... 4

Miscellaneous tools ............................................................. 6
  ExpAddField ................................................................. 6
  ExpReplaceField ............................................................ 7
  ExpResetMem ................................................................. 8
  ExpSizeOfOrdr ............................................................... 9
  ExpSizeOfRept ............................................................... 10
ExpOpenCInterface

This function opens the C-language Application Program Interface (API). You must invoke this function before you can use any other functions. If the interface is already open, the return code is set to 8.

Function prototype for ExpOpenCInterface

```
int ExpOpenCInterface(
    void
);
```

Input arguments for ExpOpenCInterface

None

Output arguments for ExpOpenCInterface

None
ExpCloseCInterface

This function closes the C-language API. Invoke this function last to end the program. If the interface is already closed, the return code is set to 8. If you use this function and there are projects left open, the return code is set to 4 and the projects are automatically closed for you.

Function prototype for ExpCloseCInterface

```c
int ExpCloseCInterface(
    void
);
```

Input arguments for ExpCloseCInterface

None

Output arguments for ExpCloseCInterface

None
Error information

Use the information in this section to work with Expedite for Windows error messages.

ExpErrorExplanation

This function retrieves information about the last error encountered. The text fields returned will not be null terminated, but will be padded on the right with blanks.

Function prototype for ExpErrorExplanation

```c
int ExpErrorExplanation(
    struct ErrorTextRspStruct * Output
);
```

Input arguments for ExpErrorExplanation

None

Output arguments for ExpErrorExplanation

```c
struct ErrorTextRspStruct
{
    /* The reason code for the error */
    char ReasonCode[EXP_MAXLEN_REASON_CODE];

    /* The brief error message */
    char ErrorMsg[EXP_MAXLEN_ERROR_MSG];

    /* The full error text */
    char ErrorTxt[EXP_MAXLEN_ERROR_TXT];

    /* The user response text */
    char Response[EXP_MAXLEN_USER_RESPONSE];
};
```
Output field definitions for ExpErrorTextRspStruct

The fields are:

**ReasonCode**
The reason code for the error.

- Max length: EXP_MAXLEN_REASON_CODE (5)
- Min length: EXP_MINLEN_REASON_CODE (1)
- Contents: Numeric data.

**ErrorMsg**
The error message text. Text will be left justified, padded on the right with blanks.

- Max length: EXP_MAXLEN_ERROR_MSG (80)
- Min length: EXP_MINLEN_ERROR_MSG (1)
- Contents: Alphanumeric data.

**ErrorTxt**
The explanation about why the error occurred. Text will be left justified, padded on the right with blanks.

- Max length: EXP_MAXLEN_ERROR_TEXT (500)
- Min length: EXP_MINLEN_ERROR_TEXT (1)
- Contents: Alphanumeric data.

**Response**
The action the user can take when an error occurs. Text will be left justified, padded on the right with blanks.

- Max length: EXP_MAXLEN_USER_RESPONSE (500)
- Min length: EXP_MINLEN_USER_RESPONSE (1)
- Contents: Alphanumeric data.
Miscellaneous tools

The functions listed in this section are tools you may find useful.

ExpAddField

This function adds the string pointed to by aSourceString to the field pointed to by aTargetField. The aSourceString value must be null terminated, and must not exceed the length of the aTargetField field. The resulting aTargetField field will be non-null terminated, padded with blanks, as required by the Expedite function calls.

Function prototype for ExpAddField

```c
void ExpAddField(
    char *aTargetField,
    char *aSourceString
);
```

Input arguments for ExpAddField

- **char *aTargetField**
  - The target field into which the source string is copied.

- **char *aSourceString**
  - The source string that is copied into the target string.

Output arguments for ExpAddField

None
Chapter 1. Getting started

Function prototype for ExpReplaceField

void ExpReplaceField(
    char *aTargetField,
    int aTargetLen,
    char * aSourceString
);

Input arguments for ExpReplaceField

char *aTargetField
    The target field into which the source string is copied.

int aTargetLen
    The maximum length of the target string.

char *aSourceString
    The source string that is copied into the target string.

Output arguments for ExpReplaceField

None
ExpResetMem

This function copies blanks (spaces) into each of the first aTargetLen bytes of the string pointed to by the aTargetMem.

Function prototype for ExpResetMem

```c
void ExpResetMem(
    char *aTargetMem,
    int aTargetLen
);
```

Input arguments for ExpResetMem

- **char *aTargetMem**
  The pointer to the start of the memory location that is set to blanks (spaces).

- **int aTargetLen**
  The amount of space allocated for the memory location.

Output arguments for ExpResetMem

None
ExpSizeOfOrdr

This function calculates the size of the order structure. The size is based on the order-identifier character specified in the anOrderType parameter. The return for this call is the length of the appropriate order-list response structure.

Function prototype for ExpSizeOfOrdr

```c
int ExpSizeOfOrdr(
    char anOrderType
);
```

Input arguments for ExpSizeOfOrdr

```c
char anOrderType
```

A character that helps identify which order to size.

- **R** Calculates the size of a receive-order structure.
- **S** Calculates the size of a send-order structure.

Output arguments for ExpSizeOfOrdr

None
ExpSizeOfRcpt

This function calculates the size of the receipt structure. The size is based on the receipt identifier character specified by the aReceiptType parameter. The return for this call is the length of the appropriate receipt list response structure.

Function prototype for ExpSizeOfRcpt

```c
int ExpSizeOfRcpt(
    char aReceiptType
);
```

Input arguments for ExpSizeOfRcpt

- **char aReceiptType**
  - A character that helps identify which receipt to size.
  
    - **R**  Calculates the size of a receive-order structure.
    - **S**  Calculates the size of a send-order structure.

Output arguments for ExpSizeOfRcpt

- None
Writing the interface for the Projects database

Using projects is a way to allow multiple users or applications (one at a time) to access the same copy of Expedite for Windows. Each project has a name, creator, and description. Each represents one user or application, or a user’s set of related tasks in Expedite for Windows. Each project has its own address book (AddressBook) database and mailbox (MailBox) database. For more information about the AddressBook database, see Chapter 3, “Writing the interface for the AddressBook database.” For more information about the MailBox database, see Chapter 4, “Writing the interface for the MailBox database.”

The database sections and the related functions discussed in this chapter are:

Projects ................................................................. 13
    ExpOpenProj ......................................................... 13
    ExpCloseProj ....................................................... 16
    ExpAddProj .......................................................... 17
    ExpAsgProjToCommProf ........................................... 24
    ExpDeleteProj ....................................................... 26
    ExpListProj .......................................................... 27
    ExpListProjOnCommProf ......................................... 32
    ExpReplaceProj ..................................................... 37
    ExpUsgProjFromCommProf ....................................... 43

Communications profiles .............................................. 45
    ExpAddCommProf .................................................. 45
    ExpDeleteCommProf ............................................... 50
    ExpListCommProf .................................................. 52
    ExpReplaceCommProf ............................................. 58
<table>
<thead>
<tr>
<th>Logon IDs</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpAddIeLogon</td>
<td>63</td>
</tr>
<tr>
<td>ExpDeleteIeLogon</td>
<td>63</td>
</tr>
<tr>
<td>ExpListIeLogon</td>
<td>68</td>
</tr>
<tr>
<td>ExpReplaceIeLogon</td>
<td>70</td>
</tr>
<tr>
<td>ExpRetrieveErrText</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>82</td>
</tr>
</tbody>
</table>
Projects

This chapter describes the functions, the input and output arguments, and the fields used in the Projects database.

ExpOpenProj

This function opens a project. If a project is already open, the return code is set to 4. If the project name specified is not found, the return code is set to 8.

Function prototype for ExpOpenProj

```c
int ExpOpenProj(
    struct ExpOpenProjReqStruct * Input,
    struct ExpProjOpenedRspStruct * Output
);
```

Input arguments for ExpOpenProj

```c
struct ExpOpenProjReqStruct
{
    /* The project password */
    char ProjPswd[EXP_MAXLEN_PJ_PROJ_PSWD];

    /* The project name*/
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```
Input field definitions for ExpOpenProjReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpProjOpenedRspStruct” on page 15.

**ProjPswd**

The project *password*. If there is no password, leave this field blank. If it matches the read password, you have read access to the project. If it matches the write password, you have read/write access. If it does not match either, an error is returned. Likewise, if there is no project password and you specify one, an error is returned.

Max length: EXP_MAXLEN_PJ_PROJ_PSWD (8)
Min length: EXP_MINLEN_PJ_PROJ_PSWD (3)
Default value: Blank
Required: No
Case sensitive: No
Validation: Yes, if a password is used on the project, it must match.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
Required: Yes
Case sensitive: No
Validation: Yes, the project must exist in the database.
Output arguments for ExpOpenProj

```c
struct ExpProjOpenedRspStruct
{
  /* The access level (read or write) */
  char AccessLevel[EXP_MAXLEN_PJ_ACCESS_LEVEL];
}
```

Output field definitions for ExpProjOpenedRspStruct

This is the output.

**AccessLevel**

The access level of the open project.

| Max length: | EXP_MAXLEN_PJ_ACCESS_LEVEL (1) |
| Min length: | EXP_MINLEN_PJ_ACCESS_LEVEL (1) |
| Contents:   | One of these valid values:      |
|             | R   Read access                   |
|             | W   Write access                  |
ExpCloseProj

This function closes a project. If the project is already closed, the return code is set to 8. If the mailbox or address book associated with this project is open, it is closed.

Function prototype for ExpCloseProj

```c
int ExpCloseProj(  
    struct ExpCloseProjReqStruct * Input  
);
```

Input arguments for ExpCloseProj

```c
struct ExpCloseProjReqStruct
{
    /* The project name */
    char ProjName [EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpCloseProjReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length**: EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length**: EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required**: Yes
- **Case Sensitive**: No
- **Validation**: Yes, the project must be open.
Chapter 2. Writing the interface for the Projects database

Function prototype for ExpAddProj

```
int ExpAddProj(
    struct ExpAddProjReqStruct * Input,
    struct ExpProjAddedRspStruct * Output
);
```

Input arguments for ExpAddProj

```
struct ExpAddProjReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The read-access password */
    char ReadPswd[EXP_MAXLEN_PJ_READ_PSWD];

    /* The write-access password */
    char WritePswd[EXP_MAXLEN_PJ_WRITE_PSWD];

    /* The retention period for receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];

    /* The drive location for projects */
    char Location[EXP_MAXLEN_PJ_LOCATION];
};
```
Input field definitions for ExpAddProjReqStruct

For a list of the output fields, see “Output field definitions for ExpProjAddedRspStruct” on page 22.

**ProjName**
The project name.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Unique value: Yes
- Required: Yes
- Case sensitive: No
- Validation: Yes, for printable data.

**Creator**
The project creator. This is usually the name of a company or a developer.

- Max length: EXP_MAXLEN_PJ_CREATOR (20)
- Min length: EXP_MINLEN_PJ_CREATOR (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**Description**
The free-format text description for the project.

- Max length: EXP_MAXLEN_PJ_DESCRIPTION (79)
- Min length: EXP_MINLEN_PJ_DESCRIPTION (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
**ReadPswd**
The password that enables read access to a project.

- Max length: EXP_MAXLEN_PJ_READ_PSWD (8)
- Min length: EXP_MINLEN_PJ_READ_PSWD (3)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**WritePswd**
The password that enables write access to a project.

- Max length: EXP_MAXLEN_PJ_WRITE_PSWD (8)
- Min length: EXP_MINLEN_PJ_WRITE_PSWD (3)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**RcpRetain**
This determines how long to keep receipts in the database.

- Max length: RM_MAXLEN_PJ_RCP_RETAIN (3)
- Min length: RM_MINLEN_PJ_RCP_RETAIN (1)
- Max value: RM_MAXVAL_PJ_RCP_RETAIN (999)
- Min value: RM_MINVAL_PJ_RCP_RETAIN (7)
- Default value: RM_DFLT_PJ_RCP_RETAIN (999)
- Required: No
- Case sensitive: No
- Validation: Yes, for numeric data.
**Location**

The drive location for the project. The default is the current drive.

- **Max length:** EXP_MAXLEN_PJ_LOCATION (1)
- **Min length:** EXP_MINLEN_PJ_LOCATION (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for printable data.
Output arguments for ExpAddProj

**struct ExpProjAddedRspStruct**

```c
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The date and time the project was last modified */
    char DateModified[EXP_MAXLEN_PJ_DATE_MODIFIED];

    /* The option for the retention of receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];

    /* The drive location for projects */
    char Location[EXP_MAXLEN_PJ_LOCATION];

    /* Reserved field */
    char Status[1];

    /* The path where the project files are located */
    char Path[EXP_MAXLEN_PJ_PATH];
};
```
Output field definitions for ExpProjAddedRspStruct

The output can include any or all of these fields.

**ProjName**
The project name.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Contents:** As entered by the user.

**Creator**
The project creator. This is usually the name of a company or a developer.

- **Max length:** EXP_MAXLEN_PJ_CREATOR (20)
- **Min length:** EXP_MINLEN_PJ_CREATOR (1)
- **Contents:** As entered by the user.

**Description**
The free-format text description for the project.

- **Max length:** EXP_MAXLEN_PJ_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_PJ_DESCRIPTION (1)
- **Contents:** As entered by the user.

**DateModified**
The date and time the project was last modified.

- **Max length:** EXP_MAXLEN_PJ_DATE_MODIFIED (14)
- **Min length:** EXP_MINLEN_PJ_DATE_MODIFIED (14)
- **Contents:** The date and time in format YYYYMMDDHHMMSS.

**RcpRetain**
The retention period for receipts.

- **Max length:** EXP_MAXLEN_PJ_RCP_RETAIN (3)
- **Min length:** EXP_MINLEN_PJ_RCP_RETAIN (1)
- **Contents:** Numeric data.
Location
The drive location for the project.

Max length: EXP_MAXLEN_PJ_LOCATION (1)
Min length: EXP_MINLEN_PJ_LOCATION (1)
Contents: A drive letter.

Status
This is a reserved field; leave it blank.
Length = 1.

Path
The drive and path where the project files are located.

Max length: EXP_MAXLEN_PJ_PATH (60)
Min length: EXP_MINLEN_PJ_PATH (1)
Contents: A file path.
ExpAsgProjToCommProf

This function assigns the project to a communications profile. If the communications profile or project name specified does not exist, the return code is set to 8. If the project is already assigned to a communications profile, the return code is set to 4.

Function prototype for ExpAsgProjToCommProf

```c
int ExpAsgProjToCommProf(
    struct ExpAsgProjToCommProfReqStruct * Input
);
```

Input arguments for ExpAsgProjToCommProf

```c
struct ExpAsgProjToCommProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];
};
```
Input field definitions for ExpAsgProjToCommProfReqStruct

There is no output associated with this function.

**ProjName**
The project name to assign to the communications profile.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: The project name must exist in the database.

**CprofName**
The communications profile name to assign to a project.

- Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
- Min length: EXP_MINLEN_CP_CPROF_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: The communications profile name must exist in the database.
ExpDeleteProj

This function deletes a project from the Project database. If the project name specified is invalid, the return code is set to 8.

If the project has communications profiles assigned to it, they are unassigned before the project is deleted.

Function prototype for ExpDeleteProj

```c
int ExpDeleteProj(
    struct ExpDeleteProjReqStruct * Input
);
```

Input arguments for ExpDeleteProj

```c
struct ExpDeleteProjReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpDeleteProjReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
ExpListProj

This function lists the projects in the Project database that correspond to the query formed in the input arguments. If the query results in no projects found, the return code is set to 4. If there is a problem with the database, the return code is set to 8.

Function prototype for ExpListProj

```c
int ExpListProj(
    struct ExpListProjReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpProjListedRspStructOutput[]
);```

Input arguments for ExpListProj

```c
struct ExpListProjReqStruct
{
    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};```
Input field definitions for ExpListProjReqStruct

For a list of the output fields, see “Output field definitions for ExpProjListedRspStruct” on page 30.

NumEntries
The number of entries to return in the output arguments.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

StartIndex
The entry number to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListProj

```c
struct ExpProjListedRspStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The date the project was last modified */
    char DateModified[EXP_MAXLEN_PJ_DATE_MODIFIED];

    /* The retention period for receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];

    /* The drive location for projects */
    char Location[EXP_MAXLEN_PJ_LOCATION];

    /* Reserved field */
    char Status[1];

    /* The path where the project files are located */
    char Path[EXP_MAXLEN_PJ_PATH];

    /* The necessity of a project password */
    char PswdNeeded[EXP_MAXLEN_PJ_PSWD_NEEDED];
};
```
Output field definitions for ExpProjListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of projects copied into the output structure.

**ProjName**
The project name.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Contents: Printable data.

**Creator**
The project creator. This is usually the name of a company or a developer.

- Max length: EXP_MAXLEN_PJ_CREATOR (20)
- Min length: EXP_MINLEN_PJ_CREATOR (1)
- Contents: As entered by the user.

**Description**
The free-format text description of the project.

- Max length: EXP_MAXLEN_PJ_DESCRIPTION (79)
- Min length: EXP_MINLEN_PJ_DESCRIPTION (1)
- Contents: As entered by the user.

**DateModified**
The date and time the project was last changed.

- Max length: EXP_MAXLEN_PJ_DATE_MODIFIED (14)
- Min length: EXP_MINLEN_PJ_DATE_MODIFIED (14)
- Contents: The date and time in YYYYMMDDHHMMSS format.
Chapter 2. Writing the interface for the Projects database

Output field definitions for ExpProjListedRspStruct

**RcpRetain**
The retention period for receipts.

- **Max length:** EXP_MAXLEN_PJ_RCP_RETAIN (3)
- **Min length:** EXP_MINLEN_PJ_RCP_RETAIN (1)
- **Max value:** EXP_MAXVAL_PJ_RCP_RETAIN (999)
- **Min value:** EXP_MINVAL_PJ_RCP_RETAIN (7)
- **Contents:** Numeric data.

**Location**
The drive location of the project.

- **Max length:** EXP_MAXLEN_PJ_LOCATION (1)
- **Min length:** EXP_MINLEN_PJ_LOCATION (1)
- **Contents:** A drive letter.

**Status**
This is a reserved field; leave it blank.
Length=1.

**Path**
The location of the project files relative to the Expedite for Windows install directory.

- **Max length:** EXP_MAXLEN_PJ_PATH (60)
- **Min length:** EXP_MINLEN_PJ_PATH (1)
- **Contents:** A file path.

**PswdNeeded**
This indicates whether a project password is needed to open the project.

- **Max length:** EXP_MAXLEN_PJ_PSWD_NEEDED (1)
- **Min length:** EXP_MINLEN_PJ_PSWD_NEEDED (1)
- **Contents:** One of these valid values:
  - **N**  A password is not needed to open this project.
  - **Y**  A password is needed to open this project.
ExpListProjOnCommProf

This function lists the projects that are assigned to a specific communications profile in the Project database. You must specify the communications profile name (CprofName). If there are no projects associated with the communications profile specified, the return code is set to 4.

Function prototype for ExpListProjOnCommProf

```c
int ExpListProjOnCommProf(
    char *CprofName,
    struct ExpListProjReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpProjListedRspStructOutput[]
);
```

Input arguments for ExpListProjOnCommProf

```c
struct ExpListProjReqStruct
{
    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number to start the query at */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```

Input field definitions for ExpListProjOnCommProfReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpListProjOnCommProfRspStruct” on page 35.

**CprofName**

The communications profile name.

- **Max length:** EXP_MAXLEN_CP_CPROF_NAME (20)
- **Min length:** EXP_MINLEN_CP_CPROF_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** The communications profile must exist.
Chapter 2. Writing the interface for the Projects database

Input field definitions for ExpListProjOnCommProfReqStruct

**NumEntries**
The number of entries to return in the output arguments.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListProjOnCommProf

```c
struct ExpProjListedRspStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The date and time the project was last modified */
    char DateModified[EXP_MAXLEN_PJ_DATE_MODIFIED];

    /* The retention period for receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];

    /* The drive location for projects */
    char Location[EXP_MAXLEN_PJ_LOCATION];

    /* Reserved field */
    char Status[1];

    /* The location of the project files */
    char Path[EXP_MAXLEN_PJ_PATH];

    /* The necessity of a project password */
    char PswdNeeded[EXP_MAXLEN_PJ_PSWD_NEEDED];
};
```
Output field definitions for ExpListProjOnCommProfRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number projects copied into the output structure.

**ProjName**
The project name.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Contents:** Printable data.

**Creator**
The project creator.

- **Max length:** EXP_MAXLEN_PJ_CREATOR (20)
- **Min length:** EXP_MINLEN_PJ_CREATOR (1)
- **Contents:** Printable data.

**Description**
The free-format text description for the project.

- **Max length:** EXP_MAXLEN_PJ_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_PJ_DESCRIPTION (1)
- **Contents:** Printable data.

**DateModified**
The date and time the project was last modified.

- **Max length:** EXP_MAXLEN_PJ_DATE_MODIFIED (14)
- **Min length:** EXP_MINLEN_PJ_DATE_MODIFIED (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.
**RcpRetain**
The retention period for receipts.

- **Max length**: EXP_MAXLEN_PJ_RCP_RETAIN (3)
- **Min length**: EXP_MINLEN_PJ_RCP_RETAIN (1)
- **Max value**: EXP_MAXVAL_PJ_RCP_RETAIN (999)
- **Min value**: EXP_MINVAL_PJ_RCP_RETAIN (7)
- **Contents**: Numeric data.

**Location**
The drive location for the project.

- **Max length**: EXP_MAXLEN_PJ_LOCATION (1)
- **Min length**: EXP_MINLEN_PJ_LOCATION (1)
- **Contents**: Printable data.

**Status**
This is a reserved field; leave it blank.
Length=1.

**Path**
The location of the project files.

- **Max length**: EXP_MAXLEN_PJ_PATH (60)
- **Min length**: EXP_MINLEN_PJ_PATH (1)
- **Contents**: A file path.

**PswdNeeded**
The necessity of a project password.

- **Max length**: EXP_MAXLEN_PJ_PSWD_NEEDED (1)
- **Min length**: EXP_MINLEN_PJ_PSWD_NEEDED (1)
- **Contents**: One of these valid values:
  - N  A password is not needed to open this project.
  - Y  A password is needed to open this project.
Chapter 2. Writing the interface for the Projects database

Function prototype for ExpReplaceProj

int ExpReplaceProj(
    struct ExpReplaceProjReqStruct * Input,
    struct ExpProjReplacedRspStruct * Output
);

Input arguments for ExpReplaceProj

struct ExpReplaceProjReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The read-access password */
    char ReadPswd[EXP_MAXLEN_PJ_READ_PSWD];

    /* The write-access password */
    char WritePswd[EXP_MAXLEN_PJ_WRITE_PSWD];

    /* The retention period for receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];
};
## Input field definitions for ExpReplaceProjReqStruct

For a list of output fields, see “Output field definitions for ExpProjReplacedRspStruct” on page 41.

### ProjName

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

### Creator

The project creator. Usually this is the name of a company or a developer.

- **Max length:** EXP_MAXLEN_PJ_CREATOR (20)
- **Min length:** EXP_MINLEN_PJ_CREATOR (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

### Description

The free-format text description for the project.

- **Max length:** EXP_MAXLEN_PJ_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_PJ_DESCRIPTION (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
ReadPswd
The new password that enables read access to a project.

- **Max length:** EXP_MAXLEN_PJ_READ_PSWD (8)
- **Min length:** EXP_MINLEN_PJ_READ_PSWD (3)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

WritePswd
The new password that enables write access to a project.

- **Max length:** EXP_MAXLEN_PJ_WRITE_PSWD (8)
- **Min length:** EXP_MINLEN_PJ_WRITE_PSWD (3)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

RcpRetain
The retention period for receipts.

- **Max length:** EXP_MAXLEN_PJ_RCP_RETAIN (3)
- **Min length:** EXP_MINLEN_PJ_RCP_RETAIN (1)
- **Max value:** EXP_MAXVAL_PJ_RCP_RETAIN (999)
- **Min value:** EXP_MINVAL_PJ_RCP_RETAIN (7)
- **Default value:** RM_DFLT_PJ_RCP_RETAIN (999)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpReplaceProj

struct ExpProjReplacedRspStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The project creator */
    char Creator[EXP_MAXLEN_PJ_CREATOR];

    /* The project description */
    char Description[EXP_MAXLEN_PJ_DESCRIPTION];

    /* The date and time the project was last modified */
    char DateModified[EXP_MAXLEN_PJ_DATE_MODIFIED];

    /* The retention period for receipts */
    char RcpRetain[EXP_MAXLEN_PJ_RCP_RETAIN];

    /* The drive location for projects */
    char Location[EXP_MAXLEN_PJ_LOCATION];

    /* Reserved field */
    char Status[1];

    /* The location of the project files */
    char Path[EXP_MAXLEN_PJ_PATH];
};
Output field definitions for ExpProjReplacedRspStruct

The output can include any or all of these fields.

**ProjName**
The project name.

- Max length: `EXP_MAXLEN_PJ_PROJ_NAME (20)`
- Min length: `EXP_MINLEN_PJ_PROJ_NAME (1)`
- Contents: As entered by the user.

**Creator**
The project creator. This is usually the name of a company or a developer.

- Max length: `EXP_MAXLEN_PJ_CREATOR (20)`
- Min length: `EXP_MINLEN_PJ_CREATOR (1)`
- Contents: As entered by the user.

**Description**
The free-format text description for the project.

- Max length: `EXP_MAXLEN_PJ_DESCRIPTION (79)`
- Min length: `EXP_MINLEN_PJ_DESCRIPTION (1)`
- Contents: As entered by the user.

**DateModified**
The date and time the project was last changed.

- Max length: `EXP_MAXLEN_PJ_DATE_MODIFIED (14)`
- Min length: `EXP_MINLEN_PJ_DATE_MODIFIED (14)`
- Contents: The date and time in format YYYYMMDDHHMMSS.

**RcpRetain**
The retention period for receipts.

- Max length: `EXP_MAXLEN_PJ_RCP_RETAIN (3)`
- Min length: `EXP_MINLEN_PJ_RCP_RETAIN (1)`
- Contents: Numeric data.
**Location**
The drive location for the project.

- **Max length:** EXP_MAXLEN_PJ_LOCATION (1)
- **Min length:** EXP_MINLEN_PJ_LOCATION (1)
- **Contents:** A drive letter.

**Status**
This is a reserved field; leave it blank.
Length=1.

**Path**
The location of the project files.

- **Max length:** EXP_MAXLEN_PJ_PATH (60)
- **Min length:** EXP_MINLEN_PJ_PATH (1)
- **Contents:** A file path.
ExpUsgProjFromCommProf

This function unassigns the project from the communications profile. If the project name or communications profile specified does not exist, the return code is set to 8. If the project is not assigned to a communications profile, the return code is set to 0.

Function prototype for ExpUsgProjFromCommProf

```c
int ExpUsgProjFromCommProf(
    struct ExpUsgProjFromCommProfReqStruct * Input
);
```

Input arguments for ExpUsgProjFromCommProf

```c
struct ExpUsgProjFromCommProfProjReqStruct
{
    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```
Input field definitions for ExpUsgProjFromCommProfReqStruct

There is no output associated with this function.

**ProjName**
The project name to unassigned from the communications profile.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: The project name must exist in the database.

**CprofName**
The communications profile name from which to unassign the project.

- Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
- Min length: EXP_MINLEN_CP_CPROF_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: The communications profile name must exist in the database.
Communications profiles

This section describes the functions, the input and output structures, and the fields used to work with communications profiles in the Projects database.

ExpAddCommProf

This function adds a new communications profile to the Project database. If the communications profile name specified already exists, the return code is set to 8.

Function prototype for ExpAddCommProf

```c
int ExpAddCommProf(
    struct ExpAddCommProfReqStruct * Input,
    struct ExpCommProfAddedRspStruct * Output
);
```

Input arguments for ExpAddCommProf

```c
struct ExpAddCommProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

    /* The communications type */
    char Commttype[EXP_MAXLEN_CP_COMMTYPE];

    /* The number of reconnects */
    char Reconnects[EXP_MAXLEN_CP_RECONNECTS];

    /* The number of redials */
    char DialCount[EXP_MAXLEN_CP_DIAL_COUNT];

    /* The dialer profile name */
    char DialProfile[EXP_MAXLEN_CP_DIAL_PROFILE];
};
```
Input field definitions for ExpAddCommProfReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpAddCommProfRspStruct” on page 48.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**CprofName**
The communications profile name.

- Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
- Min length: EXP_MINLEN_CP_CPROF_NAME (1)
- Unique: Yes, this is a key field.
- Required: Yes
- Case sensitive: No
- Validation: Yes, for printable data.

**Commttype**
The type of communications to use.

- Max length: EXP_MAXLEN_CP_COMMTYPE (1)
- Min length: EXP_MINLEN_CP_COMMTYPE (1)
- Default value: RM_DFLT_CP_COMMTYPE (D)
- Required: No
- Case sensitive: No
- Validation: The valid values are:
  - D  Use TCP/IP dial communications.
  - S  Use leased-line TCP/IP communications.
Chapter 2. Writing the interface for the Projects database

Input field definitions for ExpAddCommProfReqStruct

**Reconnects**
The number of reconnects.

- Max length: EXP_MAXLEN_CP_RECONNECTS (1)
- Min length: EXP_MINLEN_CP_RECONNECTS (1)
- Default value: RM_DFLT_CP_RECONNECTS (5)
- Required: No
- Max value: EXP_MAXVAL_CP_RECONNECTS (9)
- Min value: EXP_MINVAL_CP_RECONNECTS (0)
- Validation: Yes, as numeric data.

**DialCount**
The number of times Expedite for Windows should try each phone number when dial communications is specified.

- Max length: EXP_MAXLEN_CP_DIAL_COUNT (1)
- Min length: EXP_MINLEN_CP_DIAL_COUNT (1)
- Max value: EXP_MAXVAL_CP_DIAL_COUNT (9)
- Min value: EXP_MINVAL_CP_DIAL_COUNT (1)
- Default value: RM_DFLT_CP_DIAL_COUNT (3)
- Required: No
- Validation: Yes, for numeric data between 0 and 9.

**DialProfile**
The dial login profile name to use when dial communications is specified.

- Max length: EXP_MAXLEN_CP_DIAL_PROFILE (80)
- Min length: EXP_MINLEN_CP_DIAL_PROFILE (1)
- Default value: Blank
- Required: Yes, if CommType is D.
- Validation: No
Output arguments for ExpAddCommProf

```c
struct ExpCommProfAddedRspStruct
{
    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

    /* The communications type */
    char Commtype[EXP_MAXLEN_CP_COMMTYPE];

    /* The number of reconnects */
    char Reconnects[EXP_MAXLEN_CP_RECONNECTS];

    /* The number of redials */
    char DialCount[EXP_MAXLEN_CP_DIAL_COUNT];

    /* The dialer profile name */
    char DialProfile[EXP_MAXLEN_CP_DIAL_PROFILE];
};
```

Output field definitions for ExpAddCommProfRspStruct

The output can include any or all of these fields.

**CprofName**
The communications profile name.

Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
Min length: EXP_MINLEN_CP_CPROF_NAME (1)
Contents: Printable data.

**Commtype**
The type of communications to use.

Max length: EXP_MAXLEN_CP_COMMTYPE (1)
Min length: EXP_MINLEN_CP_COMMTYPE (1)
Contents: One of these valid values.

- **D** Use TCP/IP dial communications.
- **S** Use leased-line TCP/IP communications.
Chapter 2. Writing the interface for the Projects database

Output field definitions for ExpAddCommProfRspStruct

Reconnects
The number of reconnects.

Max length: EXP_MAXLEN_CP_RECONNECTS (1)
Min length: EXP_MINLEN_CP_RECONNECTS (1)
Contents: Numeric data.

DialCount
The number of times Expedite for Windows tried each phone number for the dial communications specified.

Max length: EXP_MAXLEN_CP_DIAL_COUNT (1)
Min length: EXP_MINLEN_CP_DIAL_COUNT (1)
Contents: Numeric data between 0 and 9.

DialProfile
The dial login profile name to use when dial communications is specified.

Max length: EXP_MAXLEN_CP_DIAL_PROFILE (80)
Min length: EXP_MINLEN_CP_DIAL_PROFILE (1)
Contents: Printable data.
ExpDeleteCommProf

This function deletes the communications profile from the Project database. If the communications profile specified does not exist, the return code is set to 4.

If there are projects assigned to the communications profile, they are unassigned before the profile is deleted.

Function prototype for ExpDeleteCommProf

```c
int ExpDeleteCommProf(
    struct ExpDeleteCommProfReqStruct * Input
);
```

Input arguments for ExpDeleteCommProf

```c
struct ExpDeleteCommProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];
};
```
Input field definitions for ExpDeleteCommProf

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**CprofName**
The communications profile name.

- Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
- Min length: EXP_MINLEN_CP_CPROF_NAME (1)
- Default value: Blank
- Unique: Yes. This is a key field.
- Case sensitive: No
- Validation: Yes, the communications profile must exist in the database.
ExpListCommProf

This function lists the communications profiles in the Project database that correspond to the query that is formed in the request structure. If the query results in no communications profiles found, the return code is set to 4.

Function prototype for ExpListCommProf

```c
int ExpListCommProf(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpCommProfListedRspStructOutput[]
);
```

Input arguments for ExpListCommProf

```c
struct ExpListReqStruct {
    /* The project name */
    char ProjName[MAXLEN_PJ_PROJ_NAME];

    /* The parameter ID to search on */
    char SearchParm[MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[MAXLEN_CF_SEARCH_VALUE];

    /* The parameter ID to sort on */
    char SortParm[MAXLEN_CF_SORT_PARM];

    /* The order in which to receive the list */
    char SortOrder[MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListCommProfReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpListCommProfRspStruct” on page 56.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** The project must be open.

**SearchParm**
The parameter ID to search on.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (25)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**SearchOp**
The search operator to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_OPER (2)
- **Min length:** EXP_MINLEN_SEARCH_OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** The valid characters are: =, <, >, <>, >=, or =<.

**SearchValue**
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, for printable data.
**SortParm**
The parameter ID sort value to use.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (25)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**SortOrder**
The order to sort the list in.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - RM_SORT_ASCENDING (A)
    Sort the list in ascending order.
  - RM_SORT_DESCENDING (D)
    Sort the list in descending order.

**NumEntries**
The number of entries to return in the output arguments.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
StartIndex
The entry number at which to start the search.

Max length: EXP_MAXLEN_CF_START_INDEX (5)
Min length: EXP_MINLEN_START_INDEX (1)
Max value: EXP_MAXVAL_START_INDEX (65535)
Min value: EXP_MINVAL_START_INDEX (1)
Required: No
Case sensitive: No
Validation: Yes, for numeric data.
Output arguments for ExpListCommProf

```c
struct ExpCommProfListedRspStruct
{
    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

    /* The communications type */
    char Commtype[EXP_MAXLEN_CP_COMMTYPE];

    /* The number of reconnects */
    char Reconnects[EXP_MAXLEN_CP_RECONNECTS];

    /* The number of redials */
    char DialCount[EXP_MAXLEN_CP_DIAL_COUNT];

    /* The dialer profile name */
    char DialProfile[EXP_MAXLEN_CP_DIAL_PROFILE];
};
```

Output field definitions for ExpListCommProfRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of communications profiles copied into the output structure.

**CprofName**
The communications profile name.

Max length: EXP_MAXLEN_CP_CPROF_NAME (20)
Min length: EXP_MINLEN_CP_CPROF_NAME (1)
Contents: Printable data.

**Commtype**
The type of communications to use.

Max length: EXP_MAXLEN_CP_COMMTYPE (1)
Min length: EXP_MINLEN_CP_COMMTYPE (1)
Contents: One of these valid values:
- D Use TCP/IP dial communications.
- S Use leased-line TCP/IP communications.
Chapter 2. Writing the interface for the Projects database

Output field definitions for ExpListCommProfRspStruct

Reconnects
The number of reconnects.

- **Max length:** EXP_MAXLEN_CP_RECONNECTS (1)
- **Min length:** EXP_MINLEN_CP_RECONNECTS (1)
- **Max value:** EXP_MAXVAL_CP_RECONNECTS (9)
- **Min value:** EXP_MINVAL_CP_RECONNECTS (0)
- **Contents:** Numeric data.

DialCount
The number of times Expedite for Windows tried each phone number for the dial communications specified.

- **Max length:** EXP_MAXLEN_CP_DIAL_COUNT (1)
- **Min length:** EXP_MINLEN_CP_DIAL_COUNT (1)
- **Contents:** Numeric data between 0 and 9.

DialProfile
The dial login profile name to use when dial communications is specified.

- **Max length:** EXP_MAXLEN_CP_DIAL_PROFILE (80)
- **Min length:** EXP_MINLEN_CP_DIAL_PROFILE (1)
- **Contents:** Printable data.
ExpReplaceCommProf

This function replaces information about a communications profile. If the communications profile name specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceCommProf

```c
int ExpReplaceCommProf(
    struct ExpReplaceCommProfReqStruct * Input,
    struct ExpCommProfReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceCommProf

```c
struct ExpReplaceCommProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The communications profile name */
    char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

    /* The communications type */
    char Commtype[EXP_MAXLEN_CP_COMMTYPE];

    /* The number of reconnects */
    char Reconnects[EXP_MAXLEN_CP_RECONNECTS];

    /* The number of redials */
    char DialCount[EXP_MAXLEN_CP_DIAL_COUNT];

    /* The dialer profile name */
    char DialProfile[EXP_MAXLEN_CP_DIAL_PROFILE];
};
```
Input field definitions for ExpReplaceCommProf

For a list of the output field descriptions, see “Output field definitions for ExpReplaceCommProfRspStruct” on page 61.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Commttype**
The type of communications to use.

- **Max length:** EXP_MAXLEN_CP_COMMTYPE (1)
- **Min length:** EXP_MINLEN_CP_COMMTYPE (1)
- **Default value:** RM_DFLT_CP_COMMTYPE (D)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - D Use TCP/IP dial communications.
  - S Use leased-line TCP/IP communications.

**Reconnects**
The number of reconnects.

- **Max length:** EXP_MAXLEN_CP_RECONNECTS (1)
- **Min length:** EXP_MINLEN_CP_RECONNECTS (1)
- **Default value:** RM_DFLT_CP_RECONNECTS (5)
- **Max value:** RM_MAXVAL_CP_RECONNECTS (9)
- **Min value:** EXP_MINVAL_CP_RECONNECTS (0)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, as numeric data.
**CprofName**
The communications profile name.

- **Max length:** EXP_MAXLEN_CP_CPROF_NAME (20)
- **Min length:** EXP_MINLEN_CP_CPROF_NAME (1)
- **Unique:** Yes, because this is a key field.
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the communications profile must exist in the database.

**DialCount**
The number of times Expedite for Windows should try each phone number when dial communications is specified.

- **Max length:** EXP_MAXLEN_CP_DIAL_COUNT (1)
- **Min length:** EXP_MINLEN_CP_DIAL_COUNT (1)
- **Min value:** EXP_MINVAL_CP_DIAL_COUNT (1)
- **Max value:** EXP_MAXVAL_CP_DIAL_COUNT (9)
- **Default value:** RM_DFLT_CP_DIAL_COUNT (3)
- **Required:** No
- **Validation:** Yes, for numeric characters between 0 and 9.

**DialProfile**
The dial login profile name to use when dial communications is specified.

- **Max length:** EXP_MAXLEN_CP_DIAL_PROFILE (80)
- **Min length:** EXP_MINLEN_CP_DIAL_PROFILE (1)
- **Default value:** Blank
- **Required:** Yes, if CommType is D.
- **Validation:** No
Chapter 2. Writing the interface for the Projects database

Output arguments for ExpReplaceCommProf

```c
struct ExpCommProfReplacedRspStruct
{
/* The communications profile name */
char CprofName[EXP_MAXLEN_CP_CPROF_NAME];

/* The communications type */
char Commttype[EXP_MAXLEN_CP_COMMTYPE];

/* The number of reconnects */
char Reconnects[EXP_MAXLEN_CP_RECONNECTS];

/* The number of redials */
char DialCount[EXP_MAXLEN_CP_DIAL_COUNT];

/* The dialer profile name */
char DialProfile[EXP_MAXLEN_CP_DIAL_PROFILE];
};
```

Output field definitions for ExpReplaceCommProfRspStruct

The output can include any or all of these fields.

**CprofName**
The communications profile name.

- **Max length:** EXP_MAXLEN_CP_CPROF_NAME (20)
- **Min length:** EXP_MINLEN_CP_CPROF_NAME (1)
- **Contents:** Printable data.

**Commttype**
The type of communications to use.

- **Max length:** EXP_MAXLEN_CP_COMMTYPE (1)
- **Min length:** EXP_MINLEN_CP_COMMTYPE (1)
- **Contents:** One of these valid values:
  - **D** Use TCP/IP dial communications.
  - **S** Use leased-line TCP/IP communications.
Reconnects
The number of reconnects.

Max length: EXP_MAXLEN_CP_RECONNECTS (1)
Min length: EXP_MINLEN_CP_RECONNECTS (1)
Contents: Numeric data.

DialCount
The number of times Expedite for Windows tried each phone number for the dial communications specified.

Max length: EXP_MAXLEN_CP_DIAL_COUNT (1)
Min length: EXP_MINLEN_CP_DIAL_COUNT (1)
Contents: Numeric data between 0 and 9.

DialProfile
The dial login profile name to use when dial communications is specified.

Max length: EXP_MAXLEN_CP_DIAL_PROFILE (80)
Min length: EXP_MINLEN_CP_DIAL_PROFILE (1)
Contents: Printable data.
Logon IDs

This section describes the functions, the input and output arguments, and fields used to work with logon IDs in the Projects database.

ExpAddIeLogon

This function adds a new Information Exchange logon ID to the Project database. If the logon ID specified already exists, the return code is set to 8.

Function prototype for ExpAddIeLogon

```c
int ExpAddIeLogon(
    struct ExpAddIeLogonReqStruct * Input,
    struct ExpIeLogonAddedRspStruct * Output
);
```

Input arguments for ExpAddIeLogon

```c
struct ExpAddIeLogonReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The Information Exchange logon record name */
    char Name[EXP_MAXLEN_IE_NAME];

    /* The Information Exchange account ID and user ID */
    char Account[EXP_MAXLEN_IE_ACCOUNT];

    /* The user’s password */
    char Password[EXP_MAXLEN_IE_PASSWORD];

    /* New Information Exchange password */
    char Npassword[EXP_MAXLEN_IE_NPASSWORD];

    /* The logon ID description */
    char Description[EXP_MAXLEN_IE_DESCRIPTION];
};
```
Input field definitions for ExpAddIeLogonReqStruct

For a list of the output fields, see “Output field definitions for ExpIeLogonAddedRspStruct” on page 67.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Name**
The Information Exchange logon record name.

- **Max length:** EXP_MAXLEN_IE_NAME (10)
- **Min length:** EXP_MINLEN_IE_NAME (1)
- **Unique:** Yes
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**Account**
The Information Exchange *account ID* and *user ID*. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- **Max length:** EXP_MAXLEN_IE_ACCOUNT (16)
- **Min length:** EXP_MINLEN_IE_ACCOUNT (9)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** No
**Password**
The user's password.

Max length: EXP_MAXLEN_IE_PASSWORD (8)
Min length: EXP_MINLEN_IE_PASSWORD (3)
Required: Yes
Case sensitive: No
Validation: No

**Npassword**
A new password. Expedite for Windows will change the password the next time the user starts a session with Information Exchange.

Max length: EXP_MAXLEN_IE_NPASSWORD (8)
Min length: EXP_MINLEN_IE_NPASSWORD (5)
Required: No
Case sensitive: No
Validation: No

**Description**
The Information Exchange logon record description.

Max length: EXP_MAXLEN_IE_DESCRIPTION (79)
Min length: EXP_MINLEN_IE_DESCRIPTION (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No
Output arguments for ExpAddIeLogon

**struct ExpIeLogonAddedRspStruct**

```c
{
    /* The Information Exchange logon record name */
    char Name[EXP_MAXLEN_IE_NAME];

    /* The Information Exchange account ID and user ID */
    char Account[EXP_MAXLEN_IE_ACCOUNT];

    /* The account ID and user ID checkpoint status */
    char Status[EXP_MAXLEN_IE_STATUS];

    /* The logon ID description */
    char Description[EXP_MAXLEN_IE_DESCRIPTION];

    /* The number of dropoff boxes assigned */
    char NumAssigned[EXP_MAXLEN_IE_NUM_ASSIGNED];
};
```
Output field definitions for ExpIeLogonAddedRspStruct

The output can include any or all of these fields.

**Name**
The Information Exchange logon record name.

- **Max length:** EXP_MAXLEN_IE_NAME (10)
- **Min length:** EXP_MINLEN_IE_NAME (1)
- **Contents:** Printable data.

**Account**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_IE_ACCOUNT (16)
- **Min length:** EXP_MINLEN_IE_ACCOUNT (9)
- **Contents:** Printable data.

**Status**
The checkpoint status for this account ID and user ID record.

- **Max length:** EXP_MAXLEN_IE_STATUS (1)
- **Min length:** EXP_MINLEN_IE_STATUS (1)
- **Contents:** One of these valid values:
  - D  The logon ID is disabled.
  - E  The logon ID is enabled.

**Description**
The Information Exchange logon record description.

- **Max length:** EXP_MAXLEN_IE_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_IE_DESCRIPTION (1)
- **Contents:** Printable data.

**NumAssigned**
The number of dropoff boxes using this Information Exchange logon record.

- **Max length:** EXP_MAXLEN_IE_NUM_ASSIGNED (5)
- **Min length:** EXP_MINLEN_IE_NUM_ASSIGNED (1)
- **Contents:** Numeric data.
ExpDeleteIeLogon

  This function deletes the Information Exchange logon ID from the Project database. If the logon ID name specified does not exist, the return code is set to 4.

Function prototype for ExpDeleteIeLogon

  int ExpDeleteIeLogon(
    struct ExpDeleteIeLogonReqStruct * Input
  );

Input arguments for ExpDeleteIeLogon

struct ExpDeleteIeLogonReqStruct
{
  /* The project name */
  char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

  /* The logon record name */
  char Name[EXP_MAXLEN_IE_NAME];
};
Input field definitions for ExpDeleteIeLogonReqStruct

This function has no output fields associated with it.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**Name**
The Information Exchange logon record name.

- Max length: EXP_MAXLEN_IE_NAME (10)
- Min length: EXP_MINLEN_IE_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the logon record must exist.
ExpListIeLogon

This function lists the Information Exchange users in the project database.

Function prototype for ExpListIeLogon

```c
int ExpListIeLogon(
    struct ExpListReqStruct * Input,
    unsigned short *EntriesReturned,
    struct ExpIeLogonListedRspStructOutput[]
);```

Input arguments for ExpListIeLogon

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};```
Input field definitions for ExpListIeLogonReqStruct

For a list of the output fields, see “Output field definitions for ExpIeLogonListedRsp-Struct” on page 75.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**SearchParm**
The field on which to search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (15)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** No
- **Validation:** Yes, the search field is MailboxId.

**SearchOp**
The search operator to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH OPER (2)
- **Min length:** EXP_MINLEN_SEARCH OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** Yes, the valid characters are: =, <, >, <>, =>, or <=.
### SearchValue
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, for printable data.

### SortParm
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** No
- **Validation:** Yes, the sort field is MailboxId.

### SortOrder
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - `RM_SORT_ASCENDING (A)`
    - Sort the list in ascending order.
  - `RM_SORT_DESCENDING (D)`
    - Sort the list in descending order.
Chapter 2. Writing the interface for the Projects database

Input field definitions for ExpListIeLogonReqStruct

**NumEntries**
The number of entries to return in the output arguments.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListIeLogon

```c
struct ExpIeLogonListedRspStruct
{
    /* The logon record name */
    char Name[EXP_MAXLEN_IE_NAME];

    /* The Information Exchange account ID and user ID */
    char Account[EXP_MAXLEN_IE_ACCOUNT];

    /* The account ID and user ID checkpoint status */
    char Status[EXP_MAXLEN_IE_STATUS];

    /* The logon record description */
    char Description[EXP_MAXLEN_IE_DESCRIPTION];

    /* The number of dropoff boxes assigned */
    char NumAssigned[EXP_MAXLEN_IE_NUM_ASSIGNED];
};
```
Output field definitions for ExpIeLogonListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of communications profiles copied into the output structure.

**Name**
The Information Exchange logon record name.

- Max length: EXP_MAXLEN_IE_NAME (10)
- Min length: EXP_MINLEN_IE_NAME (1)
- Contents: As entered by the user.

**Account**
The Information Exchange account ID and user ID. Each value is specified in eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- Max length: EXP_MAXLEN_IE_ACCOUNT (16)
- Min length: EXP_MINLEN_IE_ACCOUNT (9)
- Contents: As entered by the user.

**Status**
The checkpoint status of this account ID and user ID record.

- Max length: EXP_MAXLEN_IE_STATUS (1)
- Min length: EXP_MINLEN_IE_STATUS (1)
- Contents: One of these valid values:
  - D  The logon ID is disabled.
  - E  The logon ID is enabled.

**Description**
The Information Exchange logon record description.

- Max length: EXP_MAXLEN_IE_DESCRIPTION (79)
- Min length: EXP_MINLEN_IE_DESCRIPTION (1)
- Contents: As entered by the user.
NumAssigned
The number of dropoff boxes assigned to this Information Exchange logon record.

Max length: EXP_MAXLEN_IE_NUM_ASSIGNED (5)
Min length: EXP_MINLEN_IE_NUM_ASSIGNED (1)
Contents: Numeric data.
ExpReplaceIeLogon

This function replaces information about an Information Exchange logon ID. If the logon ID name specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceIeLogon

```c
int ExpReplaceIeLogon(
    struct ExpReplaceIeLogonReqStruct * Input,
    struct ExpIeLogonReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceIeLogon

```c
struct ExpReplaceIeLogonReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The account name */
    char Name[EXP_MAXLEN_IE_NAME];

    /* The Information Exchange account ID and user ID */
    char Account[EXP_MAXLEN_IE_ACCOUNT];

    /* The Information Exchange user’s password */
    char Password[EXP_MAXLEN_IE_PASSWORD];

    /* The new Information Exchange password */
    char Npassword[EXP_MAXLEN_IE_NPASSWORD];

    /* The Information Exchange logon ID description */
    char Description[EXP_MAXLEN_IE_DESCRIPTION];
};
```
Input field definitions for ExpReplaceIeLogonReqStruct

For a list of output fields, see “Output field definitions for ExpIeLogonReplacedRspStruct” on page 81.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**Name**
The Information Exchange logon record name.

- Max length: EXP_MAXLEN_IE_NAME (10)
- Min length: EXP_MINLEN_IE_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, for printable data.

**Account**
The Information Exchange account ID and user ID. Each value is specified in eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- Max length: EXP_MAXLEN_IE_ACCOUNT (16)
- Min length: EXP_MINLEN_IE.Account (9)
- Required: Yes
- Case sensitive: No
- Validation: Yes, for printable data.
Password
The Information Exchange logon password, which is stored as encrypted data.

- Max length: EXP_MAXLEN_IE_PASSWORD (8)
- Min length: EXP_MINLEN_IE_PASSWORD (5)
- Required: Yes
- Case sensitive: No
- Validation: No

Description
The Information Exchange logon record description.

- Max length: EXP_MAXLEN_IE_DESCRIPTION (79)
- Min length: EXP_MINLEN_IE_DESCRIPTION (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: Yes, for printable data.
Output arguments for ExpReplaceIeLogon

```c
struct ExpIeLogonReplacedRspStruct {
    /* The logon record name */
    char Name[EXP_MAXLEN_IE_NAME];

    /* The Information Exchange account ID and user ID */
    char Account[EXP_MAXLEN_IE_ACCOUNT];

    /* The account ID and user ID checkpoint status */
    char Status[EXP_MAXLEN_IE_STATUS];

    /* The logon record description */
    char Description[EXP_MAXLEN_IE_DESCRIPTION];

    /* The number of assigned dropoff boxes */
    char NumAssigned[EXP_MAXLEN_IE_NUM_ASSIGNED];
};
```
Output field definitions for ExpIeLogonReplacedRspStruct

The output can include any or all of these fields.

**Name**
The Information Exchange logon record name.

Max length: EXP_MAXLEN_IE_NAME (10)
Min length: EXP_MINLEN_IE_NAME (1)
Contents: As entered by the user.

**Account**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

Max length: EXP_MAXLEN_IE_ACCOUNT (16)
Min length: EXP_MINLEN_IE_ACCOUNT (9)
Contents: As entered by the user.

**Status**
The account ID and user ID record checkpoint status.

Max length: EXP_MAXLEN_IE_STATUS (1)
Min length: EXP_MINLEN_IE_STATUS (1)
Contents: One of these valid values:
D  The logon ID is disabled.
E  The logon ID is enabled.

**Description**
The Information Exchange logon record description.

Max length: EXP_MAXLEN_IE_DESCRIPTION (79)
Min length: EXP_MINLEN_IE_DESCRIPTION (1)
Contents: As entered by the user.

**NumAssigned**
The number of dropoff boxes that use this Information Exchange logon record.

Max length: EXP_MAXLEN_IE_NUM_ASSIGNED (5)
Min length: EXP_MINLEN_IE_NUM_ASSIGNED (1)
Contents: Numeric data.
ExpRetrieveErrText

This function gets information about an error that occurred while processing an order. Expedite for Windows will store the information if the return code is non-zero on an order receipt. The text returned is non-null terminated, left justified, and padded on the right with blanks.

Function prototype for ExpRetrieveErrText

```c
int ExpRetrieveErrText(
    struct ExpRetrieveErrTextReqStruct * Input,
    struct ExpErrTextRetrievedRspStruct * Output
);
```

Input arguments for ExpRetrieveErrText

```c
struct ExpRetrieveErrTextReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The error code */
    char ErrCode[EXP_MAXLEN_ET_ERR_CODE];

    /* The record name */
    char Name[EXP_MAXLEN_ET_NAME];

    /* Reserved field */
    char Reserved1[20];
};
```
Input field definitions for ExpRetrieveErrTextReqStruct

For a list of the output fields, see “Output field definitions for ExpErrTextRetrievedRspStruct” on page 84.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**ErrCode**
The error code indicated on the order receipt.

- **Max length:** EXP_MAXLEN_ET_ERR_CODE (5)
- **Min length:** EXP_MINLEN_ET_ERR_CODE (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Must be numeric. The error code must exist.

**Name**
The order receipt name.

- **Max length:** EXP_MAXLEN_ET_NAME (20)
- **Min length:** EXP_MINLEN_ET_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** The receipt must exist.

**Reserved**
This is a reserved field; leave it blank.

Length = 1.
Output arguments for ExpRetrieveErrText

```c
struct ExpErrTextRetrievedRspStruct
{
  /* The error code number */
  char ErrCode[EXP_MAXLEN_ET_ERR_CODE];

  /* The error message */
  char ErrMsg[EXP_MAXLEN_ET_ERR_MSG];

  /* The error explanation */
  char ErrExpl[EXP_MAXLEN_ET_ERR_EXPL];

  /* The user response */
  char ErrUresp[EXP_MAXLEN_ET_ERR_URESP];

  /* The time and date error was recorded */
  char Date[EXP_MAXLEN_ET_DATE];
};
```

Output field definitions for ExpErrTextRetrievedRspStruct

The output can include any or all of these fields.

**ErrCode**
The error code associated with the order receipt.

- **Max length:** EXP_MAXLEN_ET_ERR_CODE (5)
- **Min length:** EXP_MINLEN_ET_ERR_CODE (1)
- **Contents:** As entered by the user.

**ErrMsg**
A brief message describing the error.

- **Max length:** EXP_MAXLEN_ET_ERR_MSG (100)
- **Min length:** EXP_MINLEN_ET_ERR_MSG (1)
- **Contents:** Text, left justified, and padded on the right with blanks.

**ErrExpl**
A detailed explanation of the error.

- **Max length:** EXP_MAXLEN_ET_ERR_EXPL (950)
- **Min length:** EXP_MINLEN_ET_ERR_EXPL (1)
- **Contents:** Text, left justified, and padded on the right with blanks.
Output field definitions for `ExpErrTextRetrievedRspStruct`

Chapter 2. Writing the interface for the Projects database

**ErrUresp**
Information intended to help the user resolve the problem.

- **Max length:** EXP_MAXLEN_ET_ERR_URESP (950)
- **Min length:** EXP_MINLEN_ET_ERR_URESP (1)
- **Contents:** Text, left justified, and padded on the right with blanks.

**Date**
The date and time the error occurred.

- **Max length:** EXP_MAXLEN_ET_DATE (14)
- **Min length:** EXP_MINLEN_ET_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.
Output field definitions for ExpErrTextRetrievedRspStruct
Writing the interface for the AddressBook database

Using the AddressBook database, you can store information about a trading partner commonly found in an address book, such as names, addresses, phone numbers, and fax numbers. You can set up nicknames for your trading partners and correlate the information.

The database sections and the related functions discussed in this chapter are:

Address book ................................................................. 89
  ExpOpenAdbk ............................................................ 89
  ExpCloseAdbk ............................................................ 90

Addresses ................................................................. 91
  ExpAddAddr .............................................................. 91
  ExpAsgAddrToDistList .............................................. 105
  ExpAsgAddrToTradProf .............................................. 107
  ExpDeleteAddr .......................................................... 109
  ExpListAddr ............................................................. 111
  ExpListAddrsOnDistList .......................................... 121
  ExpListAddrsOnTradProf .......................................... 130
  ExpReplaceAddr ....................................................... 139
  ExpUsgAddrFromDistList .......................................... 153
  ExpUsgAddrFromTradProf .......................................... 155
Distribution lists ................................................................. 157
  ExpAddDistList ............................................................. 157
  ExpAsgDistListToTradProf ............................................. 160
  ExpDeleteDistList ....................................................... 162
  ExpListDistList .......................................................... 164
  ExpListDistListsOnTradProf .......................................... 169
  ExpReplaceDistList ..................................................... 173
  ExpUsgDistListFromTradProf ........................................ 176

Trading profiles ............................................................... 178
  ExpAddTradProf .......................................................... 178
  ExpDeleteTradProf ....................................................... 189
  ExpListTradProf ........................................................ 191
  ExpReplaceTradProf .................................................... 200
Address book

This chapter describes the functions, the input and output arguments, and the fields used in the AddressBook database.

ExpOpenAdbk

This function opens an address book. If the address book for the project specified is already open, the return code is set to 4.

Function prototype for ExpOpenAdbk

```c
int ExpOpenAdbk(
    struct ExpOpenAdbkReqStruct * Input
);
```

Input arguments for ExpOpenAdbk

```c
struct ExpOpenAdbkReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpOpenAdbkReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
**ExpCloseAdbk**

This function closes an address book. If the address book is already closed, the return code is set to 8.

**Function prototype for ExpCloseAdbk**

```c
int ExpCloseAdbk(
    struct ExpCloseAdbkReqStruct * Input
);
```

**Input arguments for ExpCloseAdbk**

```c
struct ExpCloseAdbkReqStruct {
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

**Input field definition for ExpCloseAdbkReqStruct**

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** `EXP_MAXLEN_PJ_PROJ_NAME` (20)
- **Min length:** `EXP_MINLEN_PJ_PROJ_NAME` (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
Addresses

These are the address functions and descriptions you use in the AddressBook database.

ExpAddAddr

This function adds a new address to the AddressBook database. If the address specified already exists, the return code is set to 8. You must specify at least one Information Exchange address for each address. The valid Information Exchange addresses are:

- IeMboxID
- IeMboxID and IeSysid
- ImxIea
- TableType, AliasTbl, and Aliasname

Function prototype for ExpAddAddr

```c
int ExpAddAddr(
    struct ExpAddAddrReqStruct * Input,
    struct ExpAddrAddedRspStruct * Output
);
```

Input arguments for ExpAddAddr

```c
struct ExpAddAddrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];

    /* The first name */
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

    /* The last name or surname */
    char Lastname[EXP_MAXLEN_AD_LASTNAME];

    /* The title */
    char Title[EXP_MAXLEN_AD_TITLE];
}```
/* The position, company, or title */
char Position[EXP_MAXLEN_AD_POSITION];

/* The organization name */
char OrgName[EXP_MAXLEN_AD_ORG_NAME];

/* The city name */
char City[EXP_MAXLEN_AD_CITY];

/* The street address */
char Street[EXP_MAXLEN_AD_STREET];

/* The state or province abbreviation */
char State[EXP_MAXLEN_AD_ST_OR_PROV];

/* The country name */
char Country[EXP_MAXLEN_AD_COUNTRY];

/* The zip code, supporting the four-digit extension */
char Zip[EXP_MAXLEN_AD_ZIP];

/* The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

/* The Information Exchange system ID */
char IeSysid[EXP_MAXLEN_AD_IE_SYSID];

/* The Information Exchange alias table type */
char TableType[EXP_MAXLEN_AD_TABLE_TYPE];

/* The alias table name */
char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];

/* The Information Exchange alias for a user’s name */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary voice telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Input field definitions for ExpAddAddrReqStruct

For a list of the output fields, see “Output field definitions for ExpAddrAddedRspStruct” on page 101.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Nickname**
The nickname for this address.

- **Max length:** EXP_MAXLEN_AD_NICKNAME (20)
- **Min length:** EXP_MINLEN_AD_NICKNAME (1)
- **Unique:** Yes
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**Firstname**
The first name or given name for this address.

- **Max length:** EXP_MAXLEN_AD_FIRSTNAME (20)
- **Min length:** EXP_MINLEN_AD_FIRSTNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Lastname**
The last name or surname for this address.

Max length: EXP_MAXLEN_AD_LASTNAME (20)
Min length: EXP_MINLEN_AD_LASTNAME (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

**Title**
The title for this address, such as Mr., Ms., Capt., or Mgr.

Max length: EXP_MAXLEN_AD_TITLE (6)
Min length: EXP_MINLEN_AD_TITLE (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

**Position**
The position for this address, such as president, programmer, or clerk.

Max length: EXP_MAXLEN_AD_POSITION (20)
Min length: EXP_MINLEN_AD_POSITION (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

**OrgName**
The organization name for this address.

Max length: EXP_MAXLEN_AD_ORG_NAME (20)
Min length: EXP_MINLEN_AD_ORG_NAME (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No


Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpAddAddrReqStruct

---

**City**

The city name for this address.

- **Max length:** EXP_MAXLEN_AD_CITY (20)
- **Min length:** EXP_MINLEN_AD_CITY (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Street**

The street number and name for this address.

- **Max length:** EXP_MAXLEN_AD_STREET (60)
- **Min length:** EXP_MINLEN_AD_STREET (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**State**

The state or province name for this address.

- **Max length:** EXP_MAXLEN_AD_ST_OR_PROV (16)
- **Min length:** EXP_MINLEN_AD_ST_OR_PROV (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Country**

The country name for this address.

- **Max length:** EXP_MAXLEN_AD_COUNTRY (20)
- **Min length:** EXP_MINLEN_AD_COUNTRY (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

---
**Zip**
The zip code or postal code for this address.

- Max length: EXP_MAXLEN_AD_ZIP (11)
- Min length: EXP_MINLEN_AD_ZIP (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMbboxId**
The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- Max length: EXP_MAXLEN_AD_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_AD_IE_MBOX_ID (9)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeSysid**
The Information Exchange system ID for this address, if different from yours. If you specify IeSysid, you must specify IeMbboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_AD_IE_SYSID (3)
- Min length: EXP_MINLEN_AD_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpAddAddrReqStruct

**ImxIea**
The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** No

**TableType**
The alias table type for this address. If you specify TableType, you must specify AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLE_TYPE (1)
- **Default value:** RM_DFLT_AD_TABLE_TYPE (P)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **G** Alias table for a global account.
  - **O** Alias table for a specific account.
  - **P** Alias table for private use.

**AliasTbl**
The Information Exchange alias table name for this address. If you specify AliasTbl, you must specify TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Aliasname**
The *alias name* for this address. If you specify Aliasname, you must specify TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_AD_ALIASNAME (16)
- **Min length:** EXP_MINLEN_AD_ALIASNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**VoicePhone1**
The primary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_1 (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**VoicePhone2**
The secondary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_2 (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**FaxNum**
The fax telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_FAX_NUM (35)
- **Min length:** EXP_MINLEN_AD_FAX_NUM (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Output arguments for ExpAddAddr

```
struct ExpAddrAddedRspStruct
{
  /* The nickname */
  char Nickname[EXP_MAXLEN_AD_NICKNAME];

  /* The first name or given name */
  char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

  /* The last name or surname */
  char Lastname[EXP_MAXLEN_AD_LASTNAME];

  /* The title */
  char Title[EXP_MAXLEN_AD_TITLE];

  /* The position, company, or title */
  char Position[EXP_MAXLEN_AD_POSITION];

  /* The organization name */
  char OrgName[EXP_MAXLEN_AD_ORG_NAME];

  /* The city name */
  char City[EXP_MAXLEN_AD_CITY];

  /* The street address */
  char Street[EXP_MAXLEN_AD_STREET];

  /* The state or province abbreviation */
  char State[EXP_MAXLEN_AD_ST_OR_PROV];

  /* The country name */
  char Country[EXP_MAXLEN_AD_COUNTRY];

  /* The zip code, supporting the four-digit extension */
  char Zip[EXP_MAXLEN_AD_ZIP];

  /* The Information Exchange account ID and user ID */
  char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

  /* The Information Exchange system ID */
  char IeSysid[EXP_MAXLEN_AD_IE_SYSID];

  /* The Information Exchange alias table type */
  char TableType[EXP_MAXLEN_AD_TABLE_TYPE];

  /* The alias table name */
  char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];
};
```
/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Output field definitions for ExpAddrAddedRspStruct

The output can include any or all of these fields.

**Nickname**
The nickname for this address.

- **Max length:** EXP_MAXLEN_AD_NICKNAME (20)
- **Min length:** EXP_MINLEN_AD_NICKNAME (1)
- **Contents:** As entered by the user.

**Firstname**
The first name or given name for this address.

- **Max length:** EXP_MAXLEN_AD_FIRSTNAME (20)
- **Min length:** EXP_MINLEN_AD_FIRSTNAME (1)
- **Contents:** As entered by the user.

**Lastname**
The last name or surname for this address.

- **Max length:** EXP_MAXLEN_AD_LASTNAME (20)
- **Min length:** EXP_MINLEN_AD_LASTNAME (1)
- **Contents:** As entered by the user.

**Title**
The title for this address, such as Mr., Ms., Capt., or Mgr.

- **Max length:** EXP_MAXLEN_AD_TITLE (6)
- **Min length:** EXP_MINLEN_AD_TITLE (1)
- **Contents:** As entered by the user.

**Position**
The position for this address, such as president, programmer, or clerk.

- **Max length:** EXP_MAXLEN_ADPOSITION (20)
- **Min length:** EXP_MINLEN_ADPOSITION (1)
- **Contents:** As entered by the user.
**OrgName**
The organization name for this address.

Max length: EXP_MAXLEN_AD_ORG_NAME (20)
Min length: EXP_MINLEN_AD_ORG_NAME (1)
Contents: As entered by the user.

**City**
The city name for this address.

Max length: EXP_MAXLEN_AD_CITY (20)
Min length: EXP_MINLEN_AD_CITY (1)
Contents: As entered by the user.

**Street**
The street number and name for this address.

Max length: EXP_MAXLEN_AD_STREET (60)
Min length: EXP_MINLEN_AD_STREET (1)
Contents: As entered by the user.

**State**
The state or province name for this address.

Max length: EXP_MAXLEN_AD_ST_OR_PROV (16)
Min length: EXP_MINLEN_AD_ST_OR_PROV (1)
Contents: As entered by the user.

**Country**
The country name for this address.

Max length: EXP_MAXLEN_AD_COUNTRY (20)
Min length: EXP_MINLEN_AD_COUNTRY (1)
Contents: As entered by the user.

**Zip**
The zip code or postal code for this address.

Max length: EXP_MAXLEN_AD_ZIP (11)
Min length: EXP_MINLEN_AD_ZIP (1)
Contents: As entered by the user.
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpAddrAddedRspStruct

**IeMboxId**
The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_AD_IE_ACCTID (16)
- **Min length:** EXP_MINLEN_AD_IE_ACCTID (9)
- **Contents:** As entered by the user.

**IeSysid**
The Information Exchange system ID for this address, if different from yours. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_AD_IE_SYSID (3)
- **Min length:** EXP_MINLEN_AD_IE_SYSID (1)
- **Contents:** As entered by the user.

**ImxIea**
The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Contents:** As entered by the user.

**TableType**
The alias table type for this address. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLETYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLETYPE (1)
- **Contents:** One of these valid values:
  - **G** Alias table for a global account.
  - **O** Alias table for a specific account.
  - **P** Alias table for private use.

**AliasTbl**
The Information Exchange alias table name for this address. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Contents:** As entered by the user.
**Aliasname**
The alias name for this address. The Aliasname is specified with TableType and AliasTbl.

Max length: EXP_MAXLEN_AD_ALIASNAME (16)
Min length: EXP_MINLEN_AD_ALIASNAME (1)
Contents: As entered by the user.

**VoicePhone1**
The primary telephone number for this address.

Max length: EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
Min length: EXP_MINLEN_AD_VOICE_PHONE_1 (1)
Contents: As entered by the user.

**VoicePhone2**
The secondary telephone number for this address.

Max length: EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
Min length: EXP_MINLEN_AD_VOICE_PHONE_2 (1)
Contents: As entered by the user.

**FaxNum**
The fax telephone number for this address.

Max length: EXP_MAXLEN_AD_FAX_NUM (35)
Min length: EXP_MINLEN_AD_FAX_NUM (1)
Contents: As entered by the user.
ExpAsgAddrToDistList

This function assigns the address to the distribution list. If the address or distribution list name specified does not exist, the return code is set to 8. If the address is already assigned to the distribution list, the return code is set to 4.

Function prototype for ExpAsgAddrToDistList

```c
int ExpAsgAddrToDistList(
    struct ExpAsgAddrToDistListReqStruct * Input
);
```

Input arguments for ExpAsgAddrToDistList

```c
struct ExpAsgAddrToDistListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
};
```
Input field definitions for ExpAsgAddrToDistListReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**DistListName**
The distribution list name to assign to the address.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.

**Nickname**
The address nickname to assign to the specified distribution list.

- Max length: EXP_MAXLEN_AD_NICKNAME (20)
- Min length: EXP_MINLEN_AD_NICKNAME (1)
- Unique: Yes
- Required: Yes
- Case sensitive: No
- Validation: Yes, the nickname must exist in the database.
ExpAsgAddrToTradProf

This function assigns the address to the trading profile. If the address or trading profile name specified does not exist, the return code is set to 8. If the address is already assigned to the trading profile, the return code is set to 4.

Function prototype for ExpAsgAddrToTradProf

```c
int ExpAsgAddrToTradProf(
    struct ExpAsgAddrToTradProfReqStruct * Input
);
```

Input arguments for ExpAsgAddrToTradProf

```c
struct ExpAsgAddrToTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TProfName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
};
```
Input field definitions for ExpAsgAddrToTradProfReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**TprofName**
The trading profile name for this address.

- Max length: EXP_MAXLEN_TP_TPROF_NAME (20)
- Min length: EXP_MINLEN_TP_TPROF_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the trading profile must exist in the database.

**Nickname**
The address nickname to assign to the specified trading profile.

- Max length: EXP_MAXLEN_AD_NICKNAME (20)
- Min length: EXP_MINLEN_AD_NICKNAME (1)
- Unique: Yes
- Required: Yes
- Case sensitive: No
- Validation: Yes, the nickname must exist in the database.
Chapter 3. Writing the interface for the AddressBook database

Function prototype for ExpDeleteAddr

`ExpDeleteAddr`  
This function deletes the address from the AddressBook database. If the address name specified does not exist, the return code is set to 4.

If the address is assigned to a distribution list, it is unassigned before it is deleted. If this address was the last one assigned to the distribution list, the distribution list remains in the AddressBook database.

If the address is assigned to a trading profile, it is unassigned before it is deleted. If this address was the last one assigned to the trading profile, the trading profile remains in the AddressBook database.

Function prototype for ExpDeleteAddr

```c
int ExpDeleteAddr(
    struct ExpDeleteAddrReqStruct * Input
);`
```

Input arguments for ExpDeleteAddr

```c
struct ExpDeleteAddrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
};`
```
Input field definitions for ExpDeleteAddrReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Nickname**
The nickname assigned to this address.

- **Max length:** EXP_MAXLEN_AD_NICKNAME (20)
- **Min length:** EXP_MINLEN_AD_NICKNAME (1)
- **Unique:** Yes
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the address nickname must exist in the database.
ExpListAddr

This function lists the addresses in the Address Book database. The list corresponds to the query formed in the response structure. If the query results in no addresses found, the return code is set to 4.

Function prototype for ExpListAddr

```c
int ExpListAddr(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpAddrListedRspStruct *Output
);
```

Input arguments for ExpListAddr

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp [EXP_MAXLEN_CF_SEARCH_OPER];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpAddrListedRspStruct” on page 117.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**SearchParm**
The field on which to search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (15)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** No
- **Validation:** Yes. The search fields are:
  - Nickname
  - Lastname
  - IeMboxId
  - Position
  - City
  - State
  - Zip

**SearchOp**
The search operator character to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_OPER (2)
- **Min length:** EXP_MINLEN_SEARCH_OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** Yes, the valid characters are: =, <, >, <>, >=, or <=.
Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpListReqStruct

**SearchValue**
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, for alphanumeric data.

**SortParm**
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** No
- **Validation:** Yes. The sort fields are:
  - Nickname
  - Lastname
  - IeMboxId
  - Position
  - City
  - State
  - Zip

**SortOrder**
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **RM_SORT_ASCENDING (A)**
    Sort the list in ascending order.
  - **RM_SORT_DESCENDING (D)**
    Sort the list in descending order.
**NumEntries**  
The number of entries to return in the response structure.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_CF_NUM_ENTRIES (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_NUM_ENTRIES (1)</td>
</tr>
<tr>
<td>Max value:</td>
<td>EXP_MAXVAL_NUM_ENTRIES (65535)</td>
</tr>
<tr>
<td>Min value:</td>
<td>EXP_MINVAL_NUM_ENTRIES (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, for numeric data.</td>
</tr>
</tbody>
</table>

**StartIndex**  
The entry number at which to start the search.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_CF_START_INDEX (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_START_INDEX (1)</td>
</tr>
<tr>
<td>Max value:</td>
<td>EXP_MAXVAL_START_INDEX (65535)</td>
</tr>
<tr>
<td>Min value:</td>
<td>EXP_MINVAL_START_INDEX (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, for numeric data.</td>
</tr>
</tbody>
</table>
Output arguments for ExpListAddr

**struct ExpAddrListedRspStruct**

```c
{
    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];

    /* The first name or given name */
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

    /* The last name or surname */
    char Lastname[EXP_MAXLEN_AD_LASTNAME];

    /* The title */
    char Title[EXP_MAXLEN_AD_TITLE];

    /* The position, company, or title */
    char Position[EXP_MAXLEN_AD_POSITION];

    /* The organization or company name */
    char OrgName[EXP_MAXLEN_AD_ORG_NAME];

    /* The city name */
    char City[EXP_MAXLEN_AD_CITY];

    /* The street address */
    char Street[EXP_MAXLEN_AD_STREET];

    /* The state or province abbreviation */
    char State[EXP_MAXLEN_AD_ST_OR_PROV];

    /* The country name */
    char Country[EXP_MAXLEN_AD_COUNTRY];

    /* The zip code, supporting the four-digit extension */
    char Zip[EXP_MAXLEN_AD_ZIP];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_AD_IE_SYSID];

    /* The Information Exchange alias table type */
    char TableType[EXP_MAXLEN_AD_TABLE_TYPE];

    /* The alias table name */
    char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];
```
/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];

};
Output field definitions for ExpAddrListedRspStruct

The output can include any or all of these fields.

**Nickname**
The nickname for this address.

- **Max length:** EXP_MAXLEN_AD_NICKNAME (20)
- **Min length:** EXP_MINLEN_AD_NICKNAME (1)
- **Contents:** As entered by the user.

**Firstname**
The first name or given name for this address.

- **Max length:** EXP_MAXLEN_AD_FIRSTNAME (20)
- **Min length:** EXP_MINLEN_AD_FIRSTNAME (1)
- **Contents:** As entered by the user.

**Lastname**
The last name or surname for this address.

- **Max length:** EXP_MAXLEN_AD_LASTNAME (20)
- **Min length:** EXP_MINLEN_AD_LASTNAME (1)
- **Contents:** As entered by the user.

**Title**
The title for this address, such as Mr., Ms., Capt., or Mgr.

- **Max length:** EXP_MAXLEN_AD_TITLE (6)
- **Min length:** EXP_MINLEN_AD_TITLE (1)
- **Contents:** As entered by the user.

**Position**
The position for this address, such as president, programmer, or clerk.

- **Max length:** EXP_MAXLEN_AD_POSITION (20)
- **Min length:** EXP_MINLEN_AD_POSITION (1)
- **Contents:** As entered by the user.
Output field definitions for ExpAddrListedRspStruct

**OrgName**
The organization name for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_ORG\_NAME} (20)
- Min length: \texttt{EXP\_MINLEN\_AD\_ORG\_NAME} (1)
- Contents: As entered by the user.

**City**
The city name for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_CITY} (20)
- Min length: \texttt{EXP\_MINLEN\_AD\_CITY} (1)
- Contents: As entered by the user.

**Street**
The street number and name for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_STREET} (60)
- Min length: \texttt{EXP\_MINLEN\_AD\_STREET} (1)
- Contents: As entered by the user.

**State**
The state or province name abbreviation for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_ST\_OR\_PROV} (2)
- Min length: \texttt{EXP\_MINLEN\_AD\_ST\_OR\_PROV} (1)
- Contents: As entered by the user.

**Country**
The country name for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_COUNTRY} (20)
- Min length: \texttt{EXP\_MINLEN\_AD\_COUNTRY} (1)
- Contents: As entered by the user.

**Zip**
The zip code or postal code for this address.

- Max length: \texttt{EXP\_MAXLEN\_AD\_ZIP} (11)
- Min length: \texttt{EXP\_MINLEN\_AD\_ZIP} (1)
- Contents: As entered by the user.
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpAddrListedRspStruct

**IeMboxId**
Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_AD_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_AD_IE_MBOX_ID (9)
- **Contents:** As entered by the user.

**IeSysid**
The Information Exchange system ID for this address, if different from yours. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_AD_IE_SYSID (3)
- **Min length:** EXP_MINLEN_AD_IE_SYSID (1)
- **Contents:** As entered by the user.

**ImxIea**
The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Contents:** As entered by the user.

**TableType**
The alias table type for this address. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLE_TYPE (1)
- **Contents:** One of these valid values:
  - G Global alias table.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The Information Exchange alias table name for this address. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Contents:** As entered by the user.
### Aliasname
The alias name for this address. The Aliasname is specified with TableType and AliasTbl.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_AD_ALIASNAME (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_AD_ALIASNAME (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

### VoicePhone1
The primary telephone number for this address.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_AD_VOICE_PHONE_1 (35)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_AD_VOICE_PHONE_1 (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

### VoicePhone2
The secondary telephone number for this address.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_AD_VOICE_PHONE_2 (35)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_AD_VOICE_PHONE_2 (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

### FaxNum
The fax telephone number for this address.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_AD_FAX_NUM (35)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_AD_FAX_NUM (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

### EntriesReturned
The number of addresses copied into the output structure.
ExpListAddrsOnDistList

This function lists the addresses assigned to a specific distribution list in the AddressBook database. The list corresponds to the query formed in the request structure. If the query results in no addresses found on the specified distribution list, the return code is set to 4.

Function prototype for ExpListAddrsOnDistList

```c
int ExpListAddrsOnDistList(
    char *DistListName,
    struct ExpListItemsReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpAddrListedRspStruct *Output
);
```

Input arguments for ExpListAddrsOnDistList

```c
struct ExpListItemsReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListItemsReqStruct

For a list of the output fields, see “Output field definitions for ExpAddrListedRspStruct” on page 126.

**DistListName**
The distribution list name.

- **Max length:** EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- **Min length:** EXP_MINLEN_DL_DIST_LIST_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the distribution list must exist in the database.

**ProjName**
The project name. Because you may have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
**StartIndex**
The entry number at which to start the search.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max length:</strong></td>
<td>EXP_MAXLEN_CF_START_INDEX (5)</td>
</tr>
<tr>
<td><strong>Min length:</strong></td>
<td>EXP_MINLEN_START_INDEX (1)</td>
</tr>
<tr>
<td><strong>Max value:</strong></td>
<td>EXP_MAXVAL_START_INDEX (65535)</td>
</tr>
<tr>
<td><strong>Min value:</strong></td>
<td>EXP_MINVAL_START_INDEX (1)</td>
</tr>
<tr>
<td><strong>Required:</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Case sensitive:</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Validation:</strong></td>
<td>Yes, for numeric data.</td>
</tr>
</tbody>
</table>
Output arguments for ExpListAddrsOnDistList

```c
struct ExpAddrListedRspStruct
{
    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];

    /* The first name or given name */
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

    /* The last name or surname */
    char Lastname[EXP_MAXLEN_AD_LASTNAME];

    /* The title */
    char Title[EXP_MAXLEN_AD_TITLE];

    /* The position, company, or title */
    char Position[EXP_MAXLEN_AD_POSITION];

    /* The organization or company name */
    char OrgName[EXP_MAXLEN_AD_ORG_NAME];

    /* The city name */
    char City[EXP_MAXLEN_AD_CITY];

    /* The street address */
    char Street[EXP_MAXLEN_AD_STREET];

    /* The state or province abbreviation */
    char State[EXP_MAXLEN_AD_ST_OR_PROV];

    /* The country name */
    char Country[EXP_MAXLEN_AD_COUNTRY];

    /* The zip code, supporting the four-digit extension */
    char Zip[EXP_MAXLEN_AD_ZIP];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_AD_IE_SYSID];

    /* The Information Exchange alias table type */
    char TableType[EXP_MAXLEN_AD_TABLE_TYPE];

    /* The alias table name */
    char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];
};
```
/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/+* The primary telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/+* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/+* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Output field definitions for ExpAddrListedRspStruct

The output can include any or all of these fields.

EntriesReturned
The number of addresses copied into the output structure.

Nickname
The nickname for this address.

- Max length: EXP_MAXLEN_AD_NICKNAME (20)
- Min length: EXP_MINLEN_AD_NICKNAME (1)
- Contents: As entered by the user.

Firstname
The first name or given name for this address.

- Max length: EXP_MAXLEN_AD_FIRSTNAME (20)
- Min length: EXP_MINLEN_AD_FIRSTNAME (1)
- Contents: As entered by the user.

Lastname
The last or surname for this address.

- Max length: EXP_MAXLEN_AD_LASTNAME (20)
- Min length: EXP_MINLEN_AD_LASTNAME (1)
- Contents: As entered by the user.

Title
The title for this address, such as Mr., Ms., Capt., or Mgr.

- Max length: EXP_MAXLEN_AD_TITLE (6)
- Min length: EXP_MINLEN_AD_TITLE (1)
- Contents: As entered by the user.

Position
The position for this address, such as president, programmer, or clerk.

- Max length: EXP_MAXLEN_AD_POSITION (20)
- Min length: EXP_MINLEN_AD_POSITION (1)
- Contents: As entered by the user.
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpAddrListedRspStruct

**OrgName**
The organization name for this address.

- **Max length:** EXP_MAXLEN_AD_ORG_NAME (20)
- **Min length:** EXP_MINLEN_AD_ORG_NAME (1)
- **Contents:** As entered by the user.

**City**
The city name for this address.

- **Max length:** EXP_MAXLEN_AD_CITY (20)
- **Min length:** EXP_MINLEN_AD_CITY (1)
- **Contents:** As entered by the user.

**Street**
The street number and name for this address.

- **Max length:** EXP_MAXLEN_AD_STREET (60)
- **Min length:** EXP_MINLEN_AD_STREET (1)
- **Contents:** As entered by the user.

**State**
The state or province abbreviation for this address.

- **Max length:** EXP_MAXLEN_AD_ST_OR_PROV (16)
- **Min length:** EXP_MINLEN_AD_ST_OR_PROV (1)
- **Contents:** As entered by the user.

**Country**
The country name for this address.

- **Max length:** EXP_MAXLEN_AD_COUNTRY (20)
- **Min length:** EXP_MINLEN_AD_COUNTRY (1)
- **Contents:** As entered by the user.

**Zip**
The zip code or postal code for this address.

- **Max length:** EXP_MAXLEN_AD_ZIP (11)
- **Min length:** EXP_MINLEN_AD_ZIP (1)
- **Contents:** As entered by the user.
**IeMboxId**
The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_AD_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_AD_IE_MBOX_ID (9)
- **Contents:** As entered by the user.

**IeSysid**
The Information Exchange system ID, if different from yours. The IeSysid is specified with the IeMboxId to create an Information Exchange address.

- **Max length:** EXP_MAXLEN_AD_IE_SYSID (3)
- **Min length:** EXP_MINLEN_AD_IE_SYSID (1)
- **Contents:** As entered by the user.

**ImxIea**
The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Contents:** As entered by the user.

**TableType**
The alias table type for this address. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLE_TYPE (1)
- **Contents:** One of these valid values:
  - G Alias table for a global account.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The Information Exchange alias table name for this address. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Contents:** As entered by the user.
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpAddrListedRspStruct

**Aliasname**
The alias name for this address.

- Max length: EXP_MAXLEN_AD_ALIASNAME (16)
- Min length: EXP_MINLEN_AD_ALIASNAME (1)
- Contents: As entered by the user.

**VoicePhone1**
The primary telephone number for this address.

- Max length: EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
- Min length: EXP_MINLEN_AD_VOICE_PHONE_1 (1)
- Contents: As entered by the user.

**VoicePhone2**
The secondary telephone number for this address.

- Max length: EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
- Min length: EXP_MINLEN_AD_VOICE_PHONE_2 (1)
- Contents: As entered by the user.

**FaxNum**
The fax telephone number for this address.

- Max length: EXP_MAXLEN_AD_FAX_NUM (35)
- Min length: EXP_MINLEN_AD_FAX_NUM (1)
- Contents: As entered by the user.
ExpListAddrsOnTradProf

This function lists the addresses assigned to a specific trading profile in the AddressBook database. The list corresponds to the query formed in the request structure. If the query results in no addresses found for the trading profile specified, the return code is set to 4.

Function prototype for ExpListAddrsOnTradProf

```c
int ExpListAddrsOnTradProf(
    char *TprofName,
    struct ExpListItemsReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpAddrListedRspStruct *Output
);
```

Input arguments for ExpListAddrsOnTradProf

```c
struct ExpListItemsReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PJ_NAME];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search*/
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListItemsReqStruct

For a list of the output fields, see “Output field definitions for ExpAddrListedRspStruct” on page 135.

**TprofName**
The trading profile name.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>TprofName</td>
<td></td>
<td>EXP_MAXLEN_TP_TPROF_NAME (20)</td>
<td>EXP_MINLEN_TP_TPROF_NAME (1)</td>
<td>Yes</td>
<td>No</td>
<td>Yes, the trading profile must exist in the database.</td>
</tr>
</tbody>
</table>

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjName</td>
<td></td>
<td>EXP_MAXLEN_PJ_PROJ_NAME (20)</td>
<td>EXP_MINLEN_PJ_PROJ_NAME (1)</td>
<td>Yes</td>
<td>No</td>
<td>Yes, the project must be open.</td>
</tr>
</tbody>
</table>

**NumEntries**
The number of entries to return in the response structure.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Max value</th>
<th>Min value</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumEntries</td>
<td></td>
<td>EXP_MAXLEN_CF_NUM_ENTRIES (5)</td>
<td>EXP_MINLEN_NUM_ENTRIES (1)</td>
<td>EXP_MAXVAL_NUM_ENTRIES (65535)</td>
<td>EXP_MINVAL_NUM_ENTRIES (1)</td>
<td>Yes</td>
<td>No</td>
<td>Yes, for numeric data.</td>
</tr>
</tbody>
</table>
**StartIndex**

The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListAddrsOnTradProf

```c
struct ExpAddrListedRspStruct {
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];
    char Lastname[EXP_MAXLEN_AD_LASTNAME];
    char Title[EXP_MAXLEN_AD_TITLE];
    char Position[EXP_MAXLEN_AD_POSITION];
    char OrgName[EXP_MAXLEN_AD_ORG_NAME];
    char City[EXP_MAXLEN_AD_CITY];
    char Street[EXP_MAXLEN_AD_STREET];
    char State[EXP_MAXLEN_AD_ST_OR_PROV];
    char Country[EXP_MAXLEN_AD_COUNTRY];
    char Zip[EXP_MAXLEN_AD_ZIP];
    char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];
    char IeSysid[EXP_MAXLEN_AD_IE_SYSID];
    char TableType[EXP_MAXLEN_AD_TABLE_TYPE];
    char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];
};
```
/* The user’s Information Exchange alias */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary or first voice telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/* The secondary or alternate telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Output field definitions for ExpAddrListedRspStruct

The output can include any or all of these fields.

EntriesReturned
The number of addresses copied into the output structure.

Nickname
The nickname for this address.

Max length: EXP_MAXLEN_AD_NICKNAME (20)
Min length: EXP_MINLEN_AD_NICKNAME (1)
Contents: As entered by the user.

Firstname
The first name or given name for this address.

Max length: EXP_MAXLEN_AD_FIRSTNAME (20)
Min length: EXP_MINLEN_AD_FIRSTNAME (1)
Contents: As entered by the user.

Lastname
The last or surname for this address.

Max length: EXP_MAXLEN_AD_LASTNAME (20)
Min length: EXP_MINLEN_AD_LASTNAME (1)
Contents: As entered by the user.

Title
The title for this address, such as Mr., Ms., Capt., or Mgr.

Max length: EXP_MAXLEN_AD_TITLE (6)
Min length: EXP_MINLEN_AD_TITLE (1)
Contents: As entered by the user.

Position
The position for this address, such as president, programmer, or clerk.

Max length: EXP_MAXLEN_AD_POSITION (20)
Min length: EXP_MINLEN_AD_POSITION (1)
Contents: As entered by the user.
Output field definitions for ExpAddrListedRspStruct

**OrgName**
The organization name for this address.
- **Max length:** EXP_MAXLEN_AD_ORG_NAME (20)
- **Min length:** EXP_MINLEN_AD_ORG_NAME (1)
- **Contents:** As entered by the user.

**City**
The city name for this address.
- **Max length:** EXP_MAXLEN_AD_CITY (20)
- **Min length:** EXP_MINLEN_AD_CITY (1)
- **Contents:** As entered by the user.

**Street**
The street number and name for this address.
- **Max length:** EXP_MAXLEN_AD_STREET (60)
- **Min length:** EXP_MINLEN_AD_STREET (1)
- **Contents:** As entered by the user.

**State**
The state or province abbreviation for this address.
- **Max length:** EXP_MAXLEN_AD_ST_OR_PROV (16)
- **Min length:** EXP_MINLEN_AD_ST_OR_PROV (1)
- **Contents:** As entered by the user.

**Country**
The country name for this address.
- **Max length:** EXP_MAXLEN_AD_COUNTRY (20)
- **Min length:** EXP_MINLEN_AD_COUNTRY (1)
- **Contents:** As entered by the user.

**Zip**
The zip or postal code for this address.
- **Max length:** EXP_MAXLEN_AD_ZIP (11)
- **Min length:** EXP_MINLEN_AD_ZIP (1)
- **Contents:** As entered by the user.
**IeMboxId**

The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_AD_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_AD_IE_MBOX_ID (9)
- **Contents:** As entered by the user.

**IeSysid**

The Information Exchange system ID, if different from yours. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_AD_IE_SYSID (3)
- **Min length:** EXP_MINLEN_AD_IE_SYSID (1)
- **Contents:** As entered by the user.

**ImxIea**

The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Contents:** As entered by the user.

**TableType**

The alias table type for this address. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLE_TYPE (1)
- **Contents:** One of these valid values:
  - **G** Alias table for a global account.
  - **O** Alias table for a specific account.
  - **P** Alias table for private use.

**AliasTbl**

The Information Exchange alias table name for this address. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Contents:** As entered by the user.
Output field definitions for ExpAddrListedRspStruct

**Aliasname**
The alias name for this address.

- **Max length:** EXP_MAXLEN_AD_ALIASNAME (16)
- **Min length:** EXP_MINLEN_AD_ALIASNAME (1)
- **Contents:** As entered by the user.

**VoicePhone1**
The primary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_1 (1)
- **Contents:** As entered by the user.

**VoicePhone2**
The secondary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_2 (1)
- **Contents:** As entered by the user.

**FaxNum**
The fax telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_FAX_NUM (35)
- **Min length:** EXP_MINLEN_AD_FAX_NUM (1)
- **Contents:** As entered by the user.
ExpReplaceAddr

This function replaces information about an address. If the address specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceAddr

```c
int ExpReplaceAddr(
    struct ExpReplaceAddrReqStruct * Input,
    struct ExpAddrReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceAddr

```c
struct ExpReplaceAddrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];

    /* The first name or given name */
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

    /* The last name or surname */
    char Lastname[EXP_MAXLEN_AD_LASTNAME];

    /* The title */
    char Title[EXP_MAXLEN_AD_TITLE];

    /* The position, company, or title */
    char Position[EXP_MAXLEN_AD_POSITION];

    /* The organization or company name */
    char OrgName[EXP_MAXLEN_AD_ORG_NAME];

    /* The city name */
    char City[EXP_MAXLEN_AD_CITY];

    /* The street address */
    char Street[EXP_MAXLEN_AD_STREET];

    /* The state or province abbreviation */
    char State[EXP_MAXLEN_AD_ST_OR_PROV];
};
```
/* The country name */
char Country[EXP_MAXLEN_AD_COUNTRY];

/* The zip code, supporting the four-digit extension */
char Zip[EXP_MAXLEN_AD_ZIP];

/* The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

/* The Information Exchange system ID */
char IeSysid[EXP_MAXLEN_AD_IE_SYSID];

/* The Information Exchange alias table type */
char TableType[EXP_MAXLEN_AD_TABLE_TYPE];

/* The alias table name */
char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];

/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary telephone number */
char VoicePhone1[EXP_MAXLEN_AD_VOICE_PHONE_1];

/* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Input field definitions for ExpReplaceAddrReqStruct

For a list of the output fields, see “Output field definitions for ExpAddrReplacedRsp-Struct” on page 149.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Nickname**
The address nickname for this address.

- **Max length:** EXP_MAXLEN_AD_NICKNAME (20)
- **Min length:** EXP_MINLEN_AD_NICKNAME (1)
- **Unique:** Yes
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the nickname must exist in the database.

**Firstname**
The first name or given name for this address.

- **Max length:** EXP_MAXLEN_AD_FIRSTNAME (20)
- **Min length:** EXP_MINLEN_AD_FIRSTNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Input field definitions for ExpReplaceAddrReqStruct

**Lastname**
The last name or surname for this address.

- **Max length:** EXP_MAXLEN_AD_LASTNAME (20)
- **Min length:** EXP_MINLEN_AD_LASTNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Title**
The title for this address, such as Mr., Ms., Capt., or Mgr.

- **Max length:** EXP_MAXLEN_AD_TITLE (6)
- **Min length:** EXP_MINLEN_AD_TITLE (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Position**
The position for this address, such as president, programmer, or clerk.

- **Max length:** EXP_MAXLEN_AD_POSITION (20)
- **Min length:** EXP_MINLEN_AD_POSITION (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**OrgName**
The organization name for this address.

- **Max length:** EXP_MAXLEN_AD_ORG_NAME (20)
- **Min length:** EXP_MINLEN_AD_ORG_NAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpReplaceAddrReqStruct

City
The city name for this address.
Max length: EXP_MAXLEN_AD_CITY (20)
Min length: EXP_MINLEN_AD_CITY (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

Street
The street number and name for this address.
Max length: EXP_MAXLEN_AD_STREET (60)
Min length: EXP_MINLEN_AD_STREET (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

State
The state or province abbreviation for this address.
Max length: EXP_MAXLEN_AD_ST_OR_PROV (16)
Min length: EXP_MINLEN_AD_ST_OR_PROV (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

Country
The country name for this address.
Max length: EXP_MAXLEN_AD_COUNTRY (20)
Min length: EXP_MINLEN_AD_COUNTRY (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No
**Zip**
The zip code or postal code for this address.

- **Max length:** EXP_MAXLEN_AD_ZIP (11)
- **Min length:** EXP_MINLEN_AD_ZIP (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**IeMboxId**
The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_AD_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_AD_IE_MBOX_ID (9)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**IeSysid**
The Information Exchange system ID, if different from yours. If you specify IeSysid, you must specify IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_AD_IE_SYSID (3)
- **Min length:** EXP_MINLEN_AD_IE_SYSID (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**ImxIea**
The Mail Exchange address for this address.

- **Max length:** EXP_MAXLEN_AD_IMX_IEA (8)
- **Min length:** EXP_MINLEN_AD_IMX_IEA (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** No

**TableType**
The alias table type for this address. If you specify TableType, you must specify AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_AD_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_AD_TABLE_TYPE (1)
- **Default value:** RM_DFLT_AD_TABLE_TYPE (P)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - G Global alias table.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The Information Exchange alias table name for this address. If you specify AliasTbl, you must specify TableType and Aliasname.

- **Max length:** EXP_MAXLEN_AD_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_AD_ALIAS_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Aliasname**
The alias name for this address. If you specify Aliasname, you must specify TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_AD_ALIASNAME (16)
- **Min length:** EXP_MINLEN_AD_ALIASNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**VoicePhone1**
The primary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_1 (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**VoicePhone2**
The secondary telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
- **Min length:** EXP_MINLEN_AD_VOICE_PHONE_2 (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**FaxNum**
The fax telephone number for this address.

- **Max length:** EXP_MAXLEN_AD_FAX_NUM (35)
- **Min length:** EXP_MINLEN_AD_FAX_NUM (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Output arguments for ExpReplaceAddr

```c
struct ExpAddrReplacedRspStruct
{
    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];

    /* The first name or given name */
    char Firstname[EXP_MAXLEN_AD_FIRSTNAME];

    /* The last name or surname */
    char Lastname[EXP_MAXLEN_AD_LASTNAME];

    /* The title */
    char Title[EXP_MAXLEN_AD_TITLE];

    /* The position, company, or title */
    char Position[EXP_MAXLEN_AD_POSITION];

    /* The organization or company name */
    char OrgName[EXP_MAXLEN_AD_ORG_NAME];

    /* The city name */
    char City[EXP_MAXLEN_AD_CITY];

    /* The street number and name */
    char Street[EXP_MAXLEN_AD_STREET];

    /* The state or province abbreviation */
    char State[EXP_MAXLEN_AD_ST_OR_PROV];

    /* The country name */
    char Country[EXP_MAXLEN_AD_COUNTRY];

    /* The zip code, supporting the four-digit extension */
    char Zip[EXP_MAXLEN_AD_ZIP];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_AD_IE_MBOX_ID];

    /* The Information Exchange alias table type */
    char TableType[EXP_MAXLEN_AD_TABLE_TYPE];
};
```
/* The alias table name */
char AliasTbl[EXP_MAXLEN_AD_ALIAS_TBL];

/* The Information Exchange alias */
char Aliasname[EXP_MAXLEN_AD_ALIASNAME];

/* The primary telephone number */
char VoicePhone1[EXP_MAXLEN_ADVOICE_PHONE_1];

/* The secondary telephone number */
char VoicePhone2[EXP_MAXLEN_AD_VOICE_PHONE_2];

/* The fax telephone number */
char FaxNum[EXP_MAXLEN_AD_FAX_NUM];
};
Output field definitions for ExpAddrReplacedRspStruct

The output can include any or all of these fields.

**Nickname**
The address nickname for this address.

- Max length: EXP_MAXLEN_AD_NICKNAME (20)
- Min length: EXP_MINLEN_AD_NICKNAME (1)
- Contents: As entered by the user.

**Firstname**
The first name or given name for this address.

- Max length: EXP_MAXLEN_AD_FIRSTNAME (20)
- Min length: EXP_MINLEN_AD_FIRSTNAME (1)
- Contents: As entered by the user.

**Lastname**
The last name or surname for this address.

- Max length: EXP_MAXLEN_AD_LASTNAME (20)
- Min length: EXP_MINLEN_AD_LASTNAME (1)
- Contents: As entered by the user.

**Title**
The title for this address, such as Mr., Ms., Capt., or Mgr.

- Max length: EXP_MAXLEN_AD_TITLE (6)
- Min length: EXP_MINLEN_AD_TITLE (1)
- Contents: As entered by the user.

**Position**
The position for this address, such as president, programmer, or clerk.

- Max length: EXP_MAXLEN_AD_POSITION (20)
- Min length: EXP_MINLEN_AD_POSITION (1)
- Contents: As entered by the user.
**OrgName**
The organization name for this address.

- **Max length:** EXP_MAXLEN_AD_ORG_NAME (20)
- **Min length:** EXP_MINLEN_AD_ORG_NAME (1)
- **Contents:** As entered by the user.

**City**
The city name for this address.

- **Max length:** EXP_MAXLEN_AD_CITY (20)
- **Min length:** EXP_MINLEN_AD_CITY (1)
- **Contents:** As entered by the user.

**Street**
The street number and name for this address.

- **Max length:** EXP_MAXLEN_AD_STREET (60)
- **Min length:** EXP_MINLEN_AD_STREET (1)
- **Contents:** As entered by the user.

**State**
The state or province abbreviation for this address.

- **Max length:** EXP_MAXLEN_AD_ST_OR_PROV (2)
- **Min length:** EXP_MINLEN_AD_ST_OR_PROV (1)
- **Contents:** As entered by the user.

**Country**
The country name for this address.

- **Max length:** EXP_MAXLEN_AD_COUNTRY (20)
- **Min length:** EXP_MINLEN_AD_COUNTRY (1)
- **Contents:** As entered by the user.

**Zip**
The zip code or postal code for this address.

- **Max length:** EXP_MAXLEN_AD_ZIP (11)
- **Min length:** EXP_MINLEN_AD_ZIP (1)
- **Contents:** As entered by the user.
**IeMboxId**

The Information Exchange account ID and user ID for this address. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** \( \text{EXP\_MAXLEN\_AD\_IE\_MBOX\_ID} (16) \)
- **Min length:** \( \text{EXP\_MINLEN\_AD\_IE\_MBOX\_ID} (9) \)
- **Contents:** As entered by the user.

**IeSysid**

The Information Exchange system ID, if different from yours. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** \( \text{EXP\_MAXLEN\_AD\_IE\_SYSID} (3) \)
- **Min length:** \( \text{EXP\_MINLEN\_AD\_IE\_SYSID} (1) \)
- **Contents:** As entered by the user.

**ImxIea**

The Mail Exchange address for this address.

- **Max length:** \( \text{EXP\_MAXLEN\_AD\_IMX\_IEA} (8) \)
- **Min length:** \( \text{EXP\_MINLEN\_AD\_IMX\_IEA} (1) \)
- **Contents:** As entered by the user.

**TableType**

The alias table type for this address. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** \( \text{EXP\_MAXLEN\_AD\_TABLE\_TYPE} (1) \)
- **Min length:** \( \text{EXP\_MINLEN\_AD\_TABLE\_TYPE} (1) \)
- **Contents:** One of these valid values:
  - \( G \) Alias table for a global account.
  - \( O \) Alias table for a specific account.
  - \( P \) Alias table for private use.

**AliasTbl**

The Information Exchange alias table name for this address. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** \( \text{EXP\_MAXLEN\_AD\_ALIAS\_TBL} (3) \)
- **Min length:** \( \text{EXP\_MINLEN\_AD\_ALIAS\_TBL} (1) \)
- **Contents:** As entered by the user.
Aliasname
The alias name for this address. The Aliasname is specified with TableType and AliasTbl.

Max length: EXP_MAXLEN_AD_ALIASNAME (16)
Min length: EXP_MINLEN_AD_ALIASNAME (1)
Contents: As entered by the user.

VoicePhone1
The primary telephone number for this address.

Max length: EXP_MAXLEN_AD_VOICE_PHONE_1 (35)
Min length: EXP_MINLEN_AD_VOICE_PHONE_1 (1)
Contents: As entered by the user.

VoicePhone2
The secondary telephone number for this address.

Max length: EXP_MAXLEN_AD_VOICE_PHONE_2 (35)
Min length: EXP_MINLEN_AD_VOICE_PHONE_2 (1)
Contents: As entered by the user.

FaxNum
The fax telephone number for this address.

Max length: EXP_MAXLEN_AD_FAX_NUM (35)
Min length: EXP_MINLEN_AD_FAX_NUM (1)
Contents: As entered by the user.
ExpUsgAddrFromDistList
This function unassigns the address from the distribution list. If the address name or distribution list name specified does not exist, the return code is set to 8. If the address is not assigned to the distribution list, the return code is set to 0.

Function prototype for ExpUsgAddrFromDistList

```c
int ExpUsgAddrFromDistList(
    struct ExpUsgAddrFromDistListReqStruct * Input
);
```

Input arguments for ExpUsgAddrFromDistList

```c
struct ExpUsgAddrFromDistListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
};
```
Input field definitions for ExpUsgAddrFromDistListReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**DistListName**
The distribution list name from which to unassign the address.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.

**Nickname**
The address nickname to unassign from the specified distribution list.

- Max length: EXP_MAXLEN_AD_NICKNAME (20)
- Min length: EXP_MINLEN_AD_NICKNAME (1)
- Unique: Yes
- Required: Yes
- Case sensitive: No
- Validation: Yes, the nickname must exist in the database.
ExpUsgAddrFromTradProf

This function unassigns the address from the trading profile. If the address name or trading profile name specified does not exist, the return code is set to 8. If the address is not assigned to a trading profile, the return code is set to 0.

Function prototype for ExpUsgAddrFromTradProf

```c
int ExpUsgAddrFromTradProf(
    struct ExpUsgAddrFromTradProfReqStruct * Input
);
```

Input arguments for ExpUsgAddrFromTradProf

```c
struct ExpUsgAddrFromTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TProfName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_AD_NICKNAME];
};
```
Input field definitions for `ExpUsgAddrFromTradProfReqStruct`

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: `EXP_MAXLEN_PJ_PROJ_NAME` (20)
- Min length: `EXP_MINLEN_PJ_PROJ_NAME` (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**TprofName**
The trading profile name from which to unassign the address.

- Max length: `EXP_MAXLEN_TP_TPROF_NAME` (20)
- Min length: `EXP_MINLEN_TP_TPROF_NAME` (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the trading profile must exist in the database.

**Nickname**
The address nickname from which to unassign the specified trading profile.

- Max length: `EXP_MAXLEN_AD_NICKNAME` (20)
- Min length: `EXP_MINLEN_AD_NICKNAME` (1)
- Unique: Yes
- Required: Yes
- Case sensitive: No
- Validation: Yes, the nickname must exist in the database.
Distribution lists

These are the functions and descriptions used for the distribution list functions in the AddressBook database.

ExpAddDistList

This function adds a new distribution list to the AddressBook database. If another distribution list by the name specified already exists, the return code is set to 8.

Function prototype for ExpAddDistList

```c
int ExpAddDistList(
    struct ExpAddDistListReqStruct * Input,
    struct ExpDistListAddedRspStruct * Output
);
```

Input arguments for ExpAddDistList

```c
struct ExpAddDistListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description[EXP_MAXLEN_DL_DESCRIPTION];
};
```
Input field definitions for ExpAddDistListReqStruct

For a list of the output fields, see “Output field definitions for ExpDistListAddedRspStruct” on page 159.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
Required: Yes
Case sensitive: No
Validation: Yes, the project must be open.

**DistListName**
The distribution list name.

Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
Required: Yes
Case sensitive: No
Validation: Yes, the distribution list must exist in the database.

**Description**
The free-format description for the distribution list.

Max length: EXP_MAXLEN_DL_DESCRIPTION (79)
Min length: EXP_MINLEN_DL_DESCRIPTION (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No
Output arguments for ExpAddDistList

```c
struct ExpDistListAddedRspStruct
{
    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description[EXP_MAXLEN_DL_DESCRIPTION];
};
```

Output field definitions for ExpDistListAddedRspStruct

The output can include one or both of these fields.

**DistListName**

The distribution list name.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Contents: As entered by the user.

**Description**

The free-format description for the distribution list.

- Max length: EXP_MAXLEN_DL_DESCRIPTION (79)
- Min length: EXP_MINLEN_DL_DESCRIPTION (1)
- Contents: As entered by the user.
ExpAsgDistListToTradProf

This function assigns the distribution list to the trading profile. If the distribution list name or trading profile name specified does not exist, the return code is set to 8. If the distribution list is already assigned to the trading profile, the return code is set to 4.

Function prototype for ExpAsgDistListToTradProf

```c
int ExpAsgDistListToTradProf(
    struct ExpAsgDistListToTradProfReqStruct * Input
);
```

Input arguments for ExpAsgDistListToTradProf

```c
struct ExpAsgDistListToTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];
};
```
Input field definitions for ExpAsgDistListToTradProfReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**TprofName**
The trading profile name to which to assign the distribution list.

- Max length: EXP_MAXLEN_TP_TPROF_NAME (20)
- Min length: EXP_MINLEN_TP_TPROF_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the trading profile must exist in the database.

**DistListName**
The distribution list name to which to assign the specified trading profile.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.
ExpDeleteDistList

This function deletes the distribution list from the AddressBook database. If the distribution list name specified does not exist, the return code is set to 4.

If the distribution list is assigned to a trading profile, it is unassigned before it is deleted. If this distribution list was the last one assigned to the trading profile, the trading partner remains in the AddressBook database.

If the distribution list has addresses assigned to it, they are unassigned before the distribution list is deleted.

Function prototype for ExpDeleteDistList

```c
int ExpDeleteDistList(
   struct ExpDeleteDistListReqStruct * Input
);
```

Input arguments ExpDeleteDistList

```c
struct ExpDeleteDistListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];
};
```
Input field definitions for ExpDeleteDistListReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**DistListName**
The distribution list name.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.
ExpListDistList

This function lists the distribution lists in the AddressBook database that correspond to the query formed in the request structure. If the query results in no distribution lists found, the return code is set to 4.

Function prototype for ExpListDistList

```
int ExpListDistList(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpDistListListedRspStruct *Output
);
```

Input arguments for ExpListDistList

```
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The operator to use for the search */
    char SearchOp [EXP_MAXLEN_CF_SEARCH_OPER];

    /* The value to use for the search */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListReqStruct

For a list of the output fields, see “Output field definitions for ExpDistListListedRspStruct” on page 168.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**SearchParm**
The field on which to search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (15)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** No
- **Validation:** Yes, the search fields are:
  - Name
  - Description

**SearchOp**
The operator to use for the query.

- **Max length:** EXP_MAXLEN_CF_SEARCH OPER (2)
- **Min length:** EXP_MINLEN_SEARCH OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** Yes, the valid characters are: =, <, >, <>, >=, or =<.
**SearchValue**
The search value to use.

- **Max length:** \( \text{EXP\_MAXLEN\_CF\_SEARCH\_VALUE} \) (25)
- **Min length:** \( \text{EXP\_MINLEN\_SEARCH\_VALUE} \) (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, for alphanumeric data.

**SortParm**
The field on which to sort.

- **Max length:** \( \text{EXP\_MAXLEN\_CF\_SORT\_PARM} \) (15)
- **Min length:** \( \text{EXP\_MINLEN\_SORT\_PARM} \) (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** No
- **Validation:** Yes, the sort fields are:
  - Name
  - Description

**SortOrder**
The order in which to sort the list.

- **Max length:** \( \text{EXP\_MAXLEN\_CF\_SORT\_ORDER} \) (1)
- **Min length:** \( \text{EXP\_MINLEN\_SORT\_ORDER} \) (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **RM\_SORT\_ASCENDING** (A)
    Sort the list in ascending order.
  - **RM\_SORT\_DESCENDING** (D)
    Sort the list in descending order.
Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpListReqStruct

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListDistList

```c
struct ExpDistListListedRspStruct
{
    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description[EXP_MAXLEN_DL_DESCRIPTION];
};
```

Output field definitions for ExpDistListListedRspStruct

The output can include one or both of these fields.

**EntriesReturned**
The number of distribution lists copied into the output structure.

**DistListName**
The distribution list name.

- **Max length:** EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- **Min length:** EXP_MINLEN_DL_DIST_LIST_NAME (1)
- **Contents:** As entered by the user.

**Description**
A short description about the distribution list.

- **Max length:** EXP_MAXLEN_DL_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_DL_DESCRIPTION (1)
- **Contents:** As entered by the user.
Chapter 3. Writing the interface for the AddressBook database

Function prototype for ExpListDistListsOnTradProf

```
int ExpListDistListsOnTradProf(
    char *TprofName,
    struct ExpListItemsReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpDistListListedRspStruct *Output
);
```

Input arguments for ExpListDistListsOnTradProf

```
struct ExpListItemsReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions ExpListDistListsOnTradProfReqStruct

For a list of the output fields, see “Output field definitions for ExpDistListListedRsp-Struct” on page 172.

**TprofName**
The trading profile name.

| Max length: | EXP_MAXLEN_TP_TPROF_NAME (20) |
| Min length: | EXP_MINLEN_TP_TPROF_NAME (1) |
| Required:   | Yes                         |
| Case sensitive: | No                   |
| Validation: | Yes, the trading profile must exist in the database. |

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

| Max length: | EXP_MAXLEN_PJ_PROJ_NAME (20) |
| Min length: | EXP_MINLEN_PJ_PROJ_NAME (1) |
| Required:   | Yes                         |
| Case sensitive: | No                   |
| Validation: | Yes, the project must be open. |

**NumEntries**
The number of entries to return in the response structure.

| Max length:  | EXP_MAXLEN_CF_NUM_ENTRIES (5) |
| Min length:  | EXP_MINLEN_NUM_ENTRIES (1) |
| Max value:   | EXP_MAXVAL_NUM_ENTRIES (65535) |
| Min value:   | EXP_MINVAL_NUM_ENTRIES (1) |
| Required:    | Yes                          |
| Case sensitive: | No                   |
| Validation:  | Yes, for numeric data.     |
StartIndex
The entry number at which to start the search.

Max length: EXP_MAXLEN_CF_START_INDEX (5)
Min length: EXP_MINLEN_START_INDEX (1)
Max value: EXP_MAXVAL_START_INDEX (65535)
Min value: EXP_MINVAL_START_INDEX (1)
Required: Yes
Case sensitive: No
Validation: Yes, for numeric data.
Output arguments for ExpListDistListsOnTradProf

```c
struct ExpDistListListedRspStruct {
    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description[EXP_MAXLEN_DL_DESCRIPTION];
};
```

Output field definitions for ExpDistListListedRspStruct

The output can include one or both of these fields.

**EntriesReturned**

The number of distribution lists copied into the output structure.

**DistListName**

The distribution list name.

- **Max length:** EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- **Min length:** EXP_MINLEN_DL_DIST_LIST_NAME (1)
- **Contents:** As entered by the user.

**Description**

A short description about the distribution list.

- **Max length:** EXP_MAXLEN_DL_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_DL_DESCRIPTION (1)
- **Contents:** As entered by the user.
ExpReplaceDistList

This function replaces information on the distribution list. If the distribution list name specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceDistList

```
int ExpReplaceDistList(
    struct ExpReplaceDistListReqStruct * Input,
    struct ExpDistListReplacedRspStruct * Output
);
```

Input arguments ExpReplaceDistList

```
struct ExpReplaceDistListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description[EXP_MAXLEN_DL_DESCRIPTION];
};
```
Input field definitions for ExpReplaceDistListReqStruct

For a list of the output fields, see “Output field definitions for ExpDistListReplacedRspStruct” on page 175.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**DistListName**
The distribution list name.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.

**Description**
The free-format description for the distribution list.

- Max length: EXP_MAXLEN_DL_DESCRIPTION (79)
- Min length: EXP_MINLEN_DL_DESCRIPTION (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
Output arguments for ExpReplaceDistList

```c
struct ExpDistListReplacedRspStruct
{
    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];

    /* The distribution list description */
    char Description [EXP_MAXLEN_DL_DESCRIPTION];
};
```

Output field definitions for ExpDistListReplacedRspStruct
The output can include one or both of these fields.

**DistListName**
The distribution list name.

- Max length: EXP_MAXLEN_DL_DIST_LIST_NAME (20)
- Min length: EXP_MINLEN_DL_DIST_LIST_NAME (1)
- Contents: As entered by the user.

**Description**
The free-format description for the distribution list.

- Max length: EXP_MAXLEN_DL_DESCRIPTION (79)
- Min length: EXP_MINLEN_DL_DESCRIPTION (1)
- Contents: As entered by the user.
**ExpUsgDistListFromTradProf**

This function unassigns the distribution list from a trading profile. If the distribution list name or trading profile name specified does not exist, the return code is set to 8. If the distribution list is not yet assigned to the trading profile, the return code is set to 0.

**Function prototype for ExpUsgDistListFromTradProf**

```c
int ExpUsgDistListFromTradProf(
    struct ExpUsgDistListFromTradProfReqStruct * Input
);
```

**Input arguments for ExpUsgDistListFromTradProf**

```c
struct ExpUsgDistListFromTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TProfName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The distribution list name */
    char DistListName[EXP_MAXLEN_DL_DIST_LIST_NAME];
};
```
Input field definitions for `ExpUsgDistListFromTradProfReqStruct`

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: `EXP_MAXLEN_PJ_PROJ_NAME` (20)
- Min length: `EXP_MINLEN_PJ_PROJ_NAME` (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**TprofName**
The trading profile name from which to unassign the distribution list.

- Max length: `EXP_MAXLEN_TP_TPROF_NAME` (20)
- Min length: `EXP_MINLEN_TP_TPROF_NAME` (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the trading profile must exist in the database.

**DistListName**
The distribution list name to unassign from the specified trading profile.

- Max length: `EXP_MAXLEN_DL_DIST_LIST_NAME` (20)
- Min length: `EXP_MINLEN_DL_DIST_LIST_NAME` (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the distribution list must exist in the database.
Trading profiles

These are the functions and descriptions you use for the trading profile functions in the AddressBook database.

ExpAddTradProf

This function adds a new trading profile to the AddressBook database. If the trading profile name already exists, the return code is set to 8.

Function prototype for ExpAddTradProf

```c
int ExpAddTradProf(
    struct ExpAddTradProfReqStruct * Input,
    struct ExpTradProfAddedRspStruct * Output
);
```
Input arguments for ExpAddTradProf

```c
struct ExpAddTradProfReqStruct {
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The sending/receiving charges */
    char Charge[EXP_MAXLEN_TP_CHARGE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_TP_MSG_CLASS];

    /* The message name */
    char MsgName[EXP_MAXLEN_TP_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_TP_MSG_SEQ_NO];

    /* The acknowledgment */
    char Ack[EXP_MAXLEN_TP_ACK];

    /* The message priority */
    char Priority[EXP_MAXLEN_TP_PRIORITY];

    /* The retention period in days */
    char Retention[EXP_MAXLEN_TP_RETENTION];

    /* Reserved field */
    char Reserved1[20];
};
```
Input field definitions for ExpAddTradProfReqStruct

For a list of the output fields, see “Output field definitions for ExpTradProfAddedRspStruct” on page 186.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**TprofName**
The trading profile name.

- **Max length:** EXP_MAXLEN_TP_TPROF_NAME (20)
- **Min length:** EXP_MINLEN_TP_TPROF_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** No
**Charge**
The message charges incurred with this trading partner. This field identifies who pays the charges. The message charge specified here overrides the Information Exchange profile options and is overridden by the order.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_CHARGE (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_CHARGE (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_TP_CHARGE (3)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td>1  The receiver pays all charges.</td>
</tr>
<tr>
<td></td>
<td>2  The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.</td>
</tr>
<tr>
<td></td>
<td>3  The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.</td>
</tr>
<tr>
<td></td>
<td>4  The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.</td>
</tr>
<tr>
<td></td>
<td>5  The sender and receiver split all charges.</td>
</tr>
<tr>
<td></td>
<td>6  The sender pays all charges.</td>
</tr>
</tbody>
</table>

**MsgClass**
The *user message class* to use for this trading partner.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_MSG_CLASS (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_MSG_CLASS (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>Blank</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>No</td>
</tr>
</tbody>
</table>
**Input field definitions for ExpAddTradProfReqStruct**

**MsgName**
The message name to use.

- Max length: `EXP_MAXLEN_TP_MSG_NAME` (8)
- Min length: `EXP_MINLEN_TP_MSG_NAME` (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**MsgSeqNo**
The message sequence number for this trading partner.

- Max length: `EXP_MAXLEN_TP_MSG_SEQ_NO` (5)
- Min length: `EXP_MINLEN_TP_MSG_SEQ_NO` (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
Ack
The acknowledgment provided when communicating with this trading partner.

Max length: EXP_MAXLEN_TP_ACK (1)
Min length: EXP_MINLEN_TP_ACK (1)
Default value: RM_DFLT_TP_ACK (N)
Required: No
Case sensitive: No
Validation: The valid values are:
- A Information Exchange provides purge acknowledgments.
- B Information Exchange provides receipt and delivery acknowledgments.
- C Information Exchange provides receipt and purge acknowledgments.
- D Information Exchange provides delivery acknowledgments.
- E Information Exchange provides purge or delivery acknowledgments.
- F Information Exchange provides receipt and either purge or delivery acknowledgments.
- N Information Exchange does not generate acknowledgments.
- R Information Exchange provides receipt acknowledgments.
**Priority**
The priority to use for this trading partner.

- **Max length:** \( \text{EXP\_MAXLEN\_TP\_PRIORITY} (1) \)
- **Min length:** \( \text{EXP\_MINLEN\_TP\_PRIORITY} (1) \)
- **Default value:** \( \text{RM\_DFLT\_TP\_PRIORITY} (N) \)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **N** Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express priority. Express delivery to users who have continuous received capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Retention**
The number of days Information Exchange leaves mail in the receiver's mailbox before it is purged. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- **Max length:** \( \text{EXP\_MAXLEN\_TP\_RETENTION} (3) \)
- **Min length:** \( \text{EXP\_MINLEN\_TP\_RETENTION} (1) \)
- **Max value:** \( \text{EXP\_MAXVAL\_TP\_RETENTION} (180) \)
- **Min value:** \( \text{EXP\_MINVAL\_TP\_RETENTION} (0) \)
- **Default value:** \( \text{RM\_DFLT\_TP\_RETENTION} (0) \)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**Reserved1**
This is a reserved field; leave it blank.

**Length = 20.**
Output arguments for ExpAddTradProf

**struct ExpTradProfAddedRspStruct**

```c
{
    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The sending/receiving charges */
    char Charge[EXP_MAXLEN_TP_CHARGE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_TP_MSG_CLASS];

    /* The message name */
    char MsgName[EXP_MAXLEN_TP_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_TP_MSG_SEQ_NO];

    /* The acknowledgment */
    char Ack[EXP_MAXLEN_TP_ACK];

    /* The message priority */
    char Priority[EXP_MAXLEN_TP_PRIORITY];

    /* The retention period in days */
    char Retention[EXP_MAXLEN_TP_RETENTION];

    /* Reserved field */
    char Reserved1[20];
};
```
Output field definitions for ExpTradProfAddedRspStruct

The output can include any or all of these fields.

**TprofName**
The trading profile name.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_TPROF_NAME (20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_TPROF_NAME (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

**Charge**
The message charges incurred with this trading partner. This field identifies who pays the charges. The message charge specified here overrides the Information Exchange profile options and is overridden by the order.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_CHARGE (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_CHARGE (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td></td>
<td>1 The receiver pays all charges.</td>
</tr>
<tr>
<td></td>
<td>2 The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.</td>
</tr>
<tr>
<td></td>
<td>3 The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.</td>
</tr>
<tr>
<td></td>
<td>4 The sender and the receiver split all charges, if agreed to by the receiver. Otherwise, the sender pays all charges.</td>
</tr>
<tr>
<td></td>
<td>5 The sender and receiver split all charges.</td>
</tr>
<tr>
<td></td>
<td>6 The sender pays all charges.</td>
</tr>
</tbody>
</table>

**MsgClass**
The user message class to use for this trading partner.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_MSG_CLASS (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_MSG_CLASS (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpTradProfAddedRspStruct

**MsgName**
The message name to use.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_TP_MSG_NAME (8)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_MSG_NAME (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

**MsgSeqNo**
The message sequence number for this trading partner.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_TP_MSG_SEQ_NO (5)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_MSG_SEQ_NO (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

**Ack**
The acknowledgment to provide when communicating with this trading partner.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_TP_ACK (1)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_ACK (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td>A</td>
<td>Information Exchange provides purge acknowledgments.</td>
</tr>
<tr>
<td>B</td>
<td>Information Exchange provides receipt and delivery acknowledgments.</td>
</tr>
<tr>
<td>C</td>
<td>Information Exchange provides receipt and purge acknowledgments.</td>
</tr>
<tr>
<td>D</td>
<td>Information Exchange provides delivery acknowledgments.</td>
</tr>
<tr>
<td>E</td>
<td>Information Exchange provides purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>F</td>
<td>Information Exchange provides receipt and either purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>N</td>
<td>Information Exchange does not provide acknowledgments.</td>
</tr>
<tr>
<td>R</td>
<td>Information Exchange provides receipt acknowledgments.</td>
</tr>
</tbody>
</table>
**Priority**
The priority to use for this trading partner.

- **Max length:** EXP_MAXLEN_TP_PRIORITY (1)
- **Min length:** EXP_MINLEN_TP_PRIORITY (1)
- **Contents:** One of these valid values:
  - **N** Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Retention**
The number of days Information Exchange leaves mail in the receiver's mailbox before it is purged. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- **Max length:** EXP_MAXLEN_TP_RETENTION (3)
- **Min length:** EXP_MINLEN_TP_RETENTION (1)
- **Contents:** Numeric data.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.
ExpDeleteTradProf

This function deletes the trading profile from the AddressBook database. If the trading profile name specified does not exist, the return code is set to 4.

If the trading profile has any addresses assigned to it, they are unassigned before the trading profile is deleted. Likewise, if the trading profile has any distribution lists assigned to it, they are unassigned before the trading profile is deleted.

Function prototype for ExpDeleteTradProf

```c
int ExpDeleteTradProf(
    struct ExpDeleteTradProfReqStruct * Input
);
```

Input arguments ExpDeleteTradProf

```c
struct ExpDeleteTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];
};
```
Input field definitions for ExpDeleteTradProfReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**TprofName**
The trading profile name.

- **Max length:** EXP_MAXLEN_TP_TPROF_NAME (20)
- **Min length:** EXP_MINLEN_TP_TPROF_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the trading profile must exist in the database.
Chapter 3. Writing the interface for the AddressBook database

Function prototype for ExpListTradProf

ExpListTradProf

This function lists the trading profiles in the AddressBook database that correspond to the query formed in the request structure. If the query results in no trading profiles found, the return code is set to 4.

Function prototype for ExpListTradProf

int ExpListTradProf(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpTradProfListedRspStruct *Output
);

Input arguments for ExpListTradProf

struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The operator to use for the search */
    char SearchOp [EXP_MAXLEN_CF_SEARCH_OPER];

    /* The value to use for the search */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
Input field definitions for ExpListTradProfReqStruct

For a list of the output fields, see “Output field definitions for ExpTradProfListedRspStruct” on page 196.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_PJ_PROJ_NAME (20)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_PJ_PROJ_NAME (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, the project must be open.</td>
</tr>
</tbody>
</table>

**SearchParm**
The field on which to search.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_CF_SEARCH_PARM (15)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SEARCH_PARM (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes, if a SearchValue is specified.</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, the search field is Name.</td>
</tr>
</tbody>
</table>

**SearchOp**
The operator to use for the query.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_CF_SEARCH_OPER (2)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SEARCH_OPER (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes, if a SearchParm is specified.</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, the valid characters are: =, &lt;, &gt;, &lt;=&gt;, &gt;=, or =&lt;.</td>
</tr>
</tbody>
</table>

**SearchValue**
The search value to use.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length:</td>
<td>EXP_MAXLEN_CF_SEARCH_VALUE (25)</td>
</tr>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SEARCH_VALUE (1)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes, if a SearchParm is specified.</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, for alphanumeric data.</td>
</tr>
</tbody>
</table>
Input field definitions for ExpListTradProfReqStruct

**SortParm**
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** No
- **Validation:** Yes, the sort field is Name.

**SortOrder**
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **RM_SORT_ASCENDING (A)**
    Sort the list in ascending order.
  - **RM_SORT_DESCENDING (D)**
    Sort the list in descending order.

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
**StartIndex**

The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
### Output arguments for ExpListTradProf

```c
struct ExpTradProfListedRspStruct
{
    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The sending/receiving charges */
    char Charge[EXP_MAXLEN_TP_CHARGE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_TP_MSG_CLASS];

    /* The message name */
    char MsgName[EXP_MAXLEN_TP_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_TP_MSG_SEQ_NO];

    /* The acknowledgment */
    char Ack[EXP_MAXLEN_TP_ACK];

    /* The message priority */
    char Priority[EXP_MAXLEN_TP_PRIORITY];

    /* The retention period in days */
    char Retention[EXP_MAXLEN_TP_RETENTION];

    /* Reserved field */
    char Reserved1[20];
};
```
Output field definitions for ExpTradProfListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of trading profiles copied into the output structure.

**TprofName**
The trading profile name.

- Max length: EXP_MAXLEN_TP_TPROF_NAME (20)
- Min length: EXP_MINLEN_TP_TPROF_NAME (1)
- Contents: As entered by the user.

**Charge**
The message charges incurred with this trading partner. This field identifies who pays the charge. The message charge specified here overrides the Information Exchange profile options, and is overridden by the order.

- Max length: EXP_MAXLEN_TP_CHARGE (1)
- Min length: EXP_MINLEN_TP_CHARGE (1)
- Contents: One of these valid values:
  1. The receiver pays all charges.
  2. The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  3. The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  4. The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  5. The sender and receiver split all charges.
  6. The sender pays all charges.
Chapter 3. Writing the interface for the AddressBook database

Output field definitions for ExpTradProfListedRspStruct

**MsgClass**
The user message class used for this trading partner.

- Max length: \text{EXP\_MAXLEN\_TP\_MSG\_CLASS} (8)
- Min length: \text{EXP\_MINLEN\_TP\_MSG\_CLASS} (1)
- Contents: As entered by the user.

**MsgName**
The message name used for this trading partner.

- Max length: \text{EXP\_MAXLEN\_TP\_MSG\_NAME} (8)
- Min length: \text{EXP\_MINLEN\_TP\_MSG\_NAME} (1)
- Contents: As entered by the user.

**MsgSeqNo**
The message sequence number used for this trading partner.

- Max length: \text{EXP\_MAXLEN\_TP\_MSG\_SEQ\_NO} (5)
- Min length: \text{EXP\_MINLEN\_TP\_MSG\_SEQ\_NO} (1)
- Contents: As entered by the user.
Ack
The acknowledgment provided when communicating with this trading partner.

| Max length: | EXP_MAXLEN_TP_ACK (1) |
| Min length: | EXP_MINLEN_TP_ACK (1) |
| Contents: | One of these valid values: |
| A | Information Exchange provides purge acknowledgments. |
| B | Information Exchange provides receipt and delivery acknowledgments. |
| C | Information Exchange provides receipt and purge acknowledgments. |
| D | Information Exchange provides delivery acknowledgments. |
| E | Information Exchange provides purge or delivery acknowledgments. |
| F | Information Exchange provides receipt and either purge or delivery acknowledgments. |
| N | Information Exchange does not provide acknowledgments. |
| R | Information Exchange provides receipt acknowledgments. |

Priority
The priority to be used for this trading partner.

| Max length: | EXP_MAXLEN_TP_PRIORITY (1) |
| Min length: | EXP_MINLEN_TP_PRIORITY (1) |
| Contents: | One of these valid values: |
| N | Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out. |
| H | High priority. The file is placed at the beginning of the receiver’s queue. |
| X | Express priority. Express delivery to users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded. |
Retention
The number of days Information Exchange leaves mail in the receiver's mailbox before it is purged. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

Max length: EXP_MAXLEN_TP_RETENTION (3)
Min length: EXP_MINLEN_TP_RETENTION (1)
Contents: Numeric data.

Reserved1
This is a reserved field; leave it blank.
Length = 20.
ExpReplaceTradProf

This function replaces information in a trading profile. If the trading profile name specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceTradProf

```c
int ExpReplaceTradProf(
    struct ExpReplaceTradProfReqStruct * Input,
    struct ExpTradProfReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceTradProf

```c
struct ExpReplaceTradProfReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The sending/receiving charges */
    char Charge[EXP_MAXLEN_TP_CHARGE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_TP_MSG_CLASS];

    /* The message name */
    char MsgName[EXP_MAXLEN_TP_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_TP_MSG_SEQ_NO];

    /* The acknowledgment */
    char Ack[EXP_MAXLEN_TP_ACK];

    /* The message priority */
    char Priority[EXP_MAXLEN_TP_PRIORITY];

    /* The retention period in days */
    char Retention[EXP_MAXLEN_TP_RETENTION];

    /* Reserved field */
    char Reserved1[20];
};
```
Input field definitions for ExpReplaceTradProfReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpTrad-ProfReplacedRspStruct” on page 207.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**TprofName**
The trading profile name.

- Max length: EXP_MAXLEN_TP_TPROF_NAME (20)
- Min length: EXP_MINLEN_TP_TPROF_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the trading profile must exist in the database.
### Charge

The message charges incurred with this trading partner. This field identifies who pays the charges. The message charge specified here overrides the Information Exchange profile options and is overridden by the order.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_CHARGE (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_CHARGE (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_TP_CHARGE (3)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td>1 The receiver pays all charges.</td>
</tr>
<tr>
<td></td>
<td>2 The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.</td>
</tr>
<tr>
<td></td>
<td>3 The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.</td>
</tr>
<tr>
<td></td>
<td>4 The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.</td>
</tr>
<tr>
<td></td>
<td>5 The sender and receiver split all charges.</td>
</tr>
<tr>
<td></td>
<td>6 The sender pays all charges.</td>
</tr>
</tbody>
</table>
### MsgClass
The user message class to use for this trading partner.

- **Max length:** EXP_MAXLEN_TP_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_TP_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

### MsgName
The message name to use.

- **Max length:** EXP_MAXLEN_TP_MSG_NAME (8)
- **Min length:** EXP_MINLEN_TP_MSG_NAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

### MsgSeqNo
The message sequence number for this trading partner.

- **Max length:** EXP_MAXLEN_TP_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_TP_MSG_SEQ_NO (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Ack
The acknowledgment to provide when communicating with this trading partner.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_TP_ACK (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_TP_ACK (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_TP_ACK (N)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td>A</td>
<td>Information Exchange provides purge acknowledgments.</td>
</tr>
<tr>
<td>B</td>
<td>Information Exchange provides receipt and delivery acknowledgments.</td>
</tr>
<tr>
<td>C</td>
<td>Information Exchange provides receipt and purge acknowledgments.</td>
</tr>
<tr>
<td>D</td>
<td>Information Exchange provides delivery acknowledgments.</td>
</tr>
<tr>
<td>E</td>
<td>Information Exchange provides purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>F</td>
<td>Information Exchange provides receipt and either purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>N</td>
<td>Information Exchange does not provide acknowledgments.</td>
</tr>
<tr>
<td>R</td>
<td>Information Exchange provides receipt acknowledgments.</td>
</tr>
</tbody>
</table>
Chapter 3. Writing the interface for the AddressBook database

Input field definitions for ExpReplaceTradProfReqStruct

**Priority**  
The priority to use for this trading partner.

- **Max length:** EXP_MAXLEN_TP_PRIORITY (1)  
- **Min length:** EXP_MINLEN_TP_PRIORITY (1)  
- **Default value:** RM_DFLT_TP_PRIORITY (N)  
- **Required:** No  
- **Case sensitive:** No  
- **Validation:** The valid values are:
  - **N** Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.  
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.  
  - **X** Express priority. Express delivery to users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Retention**  
The number of days Information Exchange leaves mail in the receiver’s mailbox before it is purged. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- **Max length:** EXP_MAXLEN_TP_RETENTION (1)  
- **Min length:** EXP_MINLEN_TP_RETENTION (1)  
- **Max value:** EXP_MAXVAL_TP_RETENTION (1)  
- **Min value:** EXP_MINVAL_TP_RETENTION (1)  
- **Default value:** RM_DFLT_TP_RETENTION (0)  
- **Required:** No  
- **Case sensitive:** No  
- **Validation:** Yes, for numeric data.

**Reserved1**  
This is a reserved field; leave it blank.  
**Length = 20.**
struct ExpTradProfReplacedRspStruct
{
    /* The trading profile name */
    char TprofName[EXP_MAXLEN_TP_TPROF_NAME];

    /* The sending/receiving charges */
    char Charge[EXP_MAXLEN_TP_CHARGE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_TP_MSG_CLASS];

    /* The message name */
    char MsgName[EXP_MAXLEN_TP_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_TP_MSG_SEQ_NO];

    /* The acknowledgment */
    char Ack[EXP_MAXLEN_TP_ACK];

    /* The message priority */
    char Priority[EXP_MAXLEN_TP_PRIORITY];

    /* The retention period in days */
    char Retention[EXP_MAXLEN_TP_RETENTION];

    /* Reserved field */
    char Reserved1[20];
};
Output field definitions for ExpTradProfReplacedRspStruct

The output can include any or all of these fields.

**TprofName**
The trading profile name.

- **Max length:** EXP_MAXLEN_TP_TPROF_NAME (20)
- **Min length:** EXP_MINLEN_TP_TPROF_NAME (1)
- **Contents:** As entered by the user.

**Charge**
The message charges incurred with this trading partner. This field identifies who pays the charges. The message charge specified here overrides the Information Exchange profile options and is overridden by the order.

- **Max length:** EXP_MAXLEN_TP_CHARGE (1)
- **Min length:** EXP_MINLEN_TP_CHARGE (1)
- **Contents:** One of these valid values:
  1. The receiver pays all charges.
  2. The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  3. The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  4. The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  5. The sender and receiver split all charges.
  6. The sender pays all charges.

**MsgClass**
The user message class to use for this trading partner.

- **Max length:** EXP_MAXLEN_TP_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_TP_MSG_CLASS (1)
- **Contents:** As entered by the user.
**MsgName**
The message name to use.

- **Max length:** EXP_MAXLEN_TP_MSG_NAME (8)
- **Min length:** EXP_MINLEN_TP_MSG_NAME (1)
- **Contents:** As entered by the user.

**MsgSeqNo**
The message sequence number for this trading partner.

- **Max length:** EXP_MAXLEN_TP_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_TP_MSG_SEQ_NO (1)
- **Contents:** As entered by the user.

**Ack**
The acknowledgment to provide when communicating with this trading partner.

- **Max length:** EXP_MAXLEN_TP_ACK (1)
- **Min length:** EXP_MINLEN_TP_ACK (1)
- **Contents:** One of these valid values:
  - **A** Information Exchange provides purge acknowledgments.
  - **B** Information Exchange provides receipt and delivery acknowledgments.
  - **C** Information Exchange provides receipt and purge acknowledgments.
  - **D** Information Exchange provides delivery acknowledgments.
  - **E** Information Exchange provides purge or delivery acknowledgments.
  - **F** Information Exchange provides receipt and either purge or delivery acknowledgments.
  - **N** Information Exchange does not provide acknowledgments.
  - **R** Information Exchange provides receipt acknowledgments.
**Priority**
The priority to use for this trading partner.

- **Max length:** `EXP_MAXLEN_TP_PRIORITY (1)`
- **Min length:** `EXP_MINLEN_TP_PRIORITY (1)`
- **Contents:**
  - **N** Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express Priority. Express delivery to users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Retention**
The number of days Information Exchange leaves mail in the receiver's mailbox before it is purged. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- **Max length:** `EXP_MAXLEN_TP_RETENTION (3)`
- **Min length:** `EXP_MINLEN_TP_RETENTION (1)`
- **Contents:** Numeric data.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.
Output field definitions for ExpTradProfReplacedRspStruct
Writing the interface for the MailBox database

The MailBox database contains the objects associated with processed orders. The objects in the MailBox database are:

- The OrderShelf
  - Contains all the orders in the database.

- The Dropoff Box
  - Contains the orders for processing.

- The ReceiptShelf
  - Contains the order receipts and the session receipts.

The database sections and the related functions discussed in this chapter are:

- MailBox ................................................................. 213
  - ExpOpenMbox ...................................................... 213
  - ExpCloseMbox ..................................................... 214

- Order shelf and receipt shelf .................................... 215
  - ExpListOrdrsOnOrdrShlf ......................................... 215
  - ExpListRcptsOnSessRcpt ........................................ 221

- Dropoff boxes ...................................................... 225
  - ExpAddDropBox .................................................... 225
  - ExpDeleteDropBox ................................................ 242
  - ExpListDropBox ................................................... 244
  - ExpReplaceDropBox .............................................. 257
Receive orders .................................................................274
  ExpAddRecvOrdr .........................................................274
  ExpAsgRecvOrdrToDropBox .............................................302
  ExpDeleteRecvOrdr ......................................................304
  ExpListRecvOrdr .........................................................306
  ExpReplaceRecvOrdr .....................................................322
  ExpUsgRecvOrdrFromDropBox .........................................348

Send orders .................................................................350
  ExpAddSendOrdr ........................................................350
  ExpAsgSendOrdrToDropBox .............................................378
  ExpDeleteSendOrdr ......................................................380
  ExpListSendOrdr ........................................................382
  ExpReplaceSendOrdr ....................................................399
  ExpUsgSendOrdrFromDropBox .........................................426

Session receipts ........................................................428
  ExpDeleteSessRept .....................................................428
  ExpListSessRept .........................................................430

Receive receipts ........................................................439
  ExpListRecvRept .......................................................439

Send receipts ...........................................................460
  ExpListSendRept .......................................................460
Chapter 4. Writing the interface for the MailBox database

Function prototype for ExpOpenMbox

This chapter describes the functions, the input and output arguments, and the field descriptions you use to work in the MailBox database.

ExpOpenMbbox

This function opens the mailbox.

Function prototype for ExpOpenMbbox

```c
int ExpOpenMbbox(
    struct ExpOpenMbboxReqStruct * Input,
);
```

Input arguments for ExpOpenMbbox

```c
struct ExpOpenMbboxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpOpenMbboxReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
ExpCloseMbox

This function closes the mailbox.

Function prototype for ExpCloseMbox

```c
int ExpCloseMbox(
    struct ExpCloseMboxReqStruct * Input
);
```

Input arguments for ExpCloseMbox

```c
struct ExpCloseMboxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpCloseMboxReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
Order shelf and receipt shelf

These are the functions, the input and output arguments, and the field descriptions used to work with the order shelf and the receipt shelf in the MailBox database.

ExpListOrdrsOnOrdrShlf

This function lists the orders on the order shelf in the Mailbox database that correspond to the query formed in the request structure. If the query results in no orders found on the order shelf, the return code is set to 4.

Function prototype for ExpListOrdrsOnOrdrShlf

```c
int ExpListOrdrsOnOrdrShlf(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    char *Output
);
```
Input arguments for ExpListOrdrsOnOrdrShlf

```c
struct ExpListReqStruct
{
/* The project name */
char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

/* The field on which to search */
char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

/* The search operator to use */
char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

/* The search value to use */
char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

/* The field on which to sort */
char SortParm[EXP_MAXLEN_CF_SORT_PARM];

/* The order in which to sort the list */
char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

/* The number of entries requested for receiving */
char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

/* The entry number at which to start the search */
char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListOrdrsOnOrdrShlfReqStruct

For a list of the output fields, see page 220.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**SearchParm**
The field on which to search.

- Max length: EXP_MAXLEN_CF_SEARCH_PARM (15)
- Min length: EXP_MINLEN_SEARCH_PARM (1)
- Required: Yes, if a SearchValue is specified.
- Case sensitive: Yes
- Validation: Yes, the search fields are:
  - Name
  - Description

**SearchOp**
The search operator to use.

- Max length: EXP_MAXLEN_CF_SEARCH_OPER (2)
- Min length: EXP_MINLEN_SEARCH_OPER (1)
- Required: Yes, if a SearchParm is specified.
- Case sensitive: No
- Validation: Yes, the valid characters are: =, <, >, <=, >=, or <=.
**SearchValue**

The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** No

**SortParm**

The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the search fields are:
  - Name
  - Description

**SortOrder**

The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **RM_SORT_ASCENDING (A)**
    Sort the list in ascending order.
  - **RM_SORT_DESCENDING (D)**
    Sort the list in descending order.
**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output field definitions for ExpListOrdrsOnOrdrShlfRspStruct

The output can include any or all of these structures.

*EntriesReturned*

The number of orders copied into the output area.

*Output*

The orders obtained as a result of the list request. They can be either of the following:

*ExpRecvOrdrListedRspStruct*

This structure is described on page 310.

*ExpSendOrdrListedRspStruct*

This structure is described on page 386.
ExpListRcptsOnSessRcpt

This function lists the order receipts on a session receipt in the Mailbox database. If there are no order receipts found on the specified session receipt, the return code is set to 4.

Function prototype for ExpListRcptsOnSessRcpt

```c
int ExpListRcptsOnSessRcpt(
    char *SrcptName,
    struct ExpListItemsReqStruct *Input,
    unsigned short *EntriesReturned,
    char *Output
);
```

Input arguments for ExpListRcptsOnSessRcpt

```c
struct ExpListItemsReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListRcptsOnSessRcptReqStruct

For a list of the output field definitions, see page 224.

**SrcptName**
The session receipt name.

- **Max length:** EXP_MAXLEN_SR_SRCPT_NAME (20)
- **Min length:** EXP_MINLEN_SR_SRCPT_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the session receipt must exist in the database.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
**StartIndex**

The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output field definitions for ExpListRcptsOnSessRcptRspStruct

The output can include any or all of these structures.

**Entries Returned**

The number of orders copied into the output area.

**Output**

The receipts obtained as a result of the list request. They can be either of the following:

**ExpRecvRcptListedRspStruct**

This structure is described on page 443.

**ExpSendRcptListedRspStruct**

This structure is described on page 464.
Dropoff boxes

These are the functions, the input and output arguments, and the field descriptions you use to work with dropoff boxes in the MailBox database.

ExpAddDropBox

This function adds a new dropoff box to the MailBox database. If the dropoff box name already exists, the return code is set to 8.

Function prototype for ExpAddDropBox

```c
int ExpAddDropBox(
    struct ExpAddDropBoxReqStruct * Input,
    struct ExpDropBoxAddedRspStruct * Output
);
```

Input arguments for ExpAddDropBox

```c
struct ExpAddDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The Information Exchange logon record ID for this dropoff box */
    char IeMboxIdName[EXP_MAXLEN_DR_IE_MBOX_NAME];

    /* The option to receive e-mail at end of session */
    char EmailOpt[EXP_MAXLEN_DR_EMAIL_OPT];

    /* The option to receive system messages at end of session */
    char SysmsgOpt[EXP_MAXLEN_DR_SYSMSG_OPT];

    /* The option to remove orders from the dropoff box */
    char OrdersOpt[EXP_MAXLEN_DR_ORDERS_OPT];

    /* The option to display session status */
    char AppStatus[EXP_MAXLEN_DR_APP_STATUS];

    /* Reserved field */
    char Reserved1[1];
};
```
/* Reserved field */
char Reserved2[1];

/* The dropoff box recovery type */
char DropRecovery[EXP_MAXLEN_DR_DROP_RECOVERY];

/* The level of recovery */
char SessRecovery[EXP_MAXLEN_DR_SESS_RECOVERY];

/* The number of bytes to send between checkpoints */
char Commitdata[EXP_MAXLEN_DR_COMMITDATA];

/* The maximum number of messages received between commits */
char Maxmsgs[EXP_MAXLEN_DR_MAXMSG];

/* The size of the data segments to send */
char Msgsize[EXP_MAXLEN_DR_MSGSIZE];

/* The translation table name for a dropoff box */
char XlateTbl[EXP_MAXLEN_DR_XLATE_TBL];

/* The action to take if a file exists */
char Overwrite[EXP_MAXLEN_DR_OVERWRITE];

/* The time zone */
char TimeZone[EXP_MAXLEN_DR_TIME_ZONE];

};
Input field definitions for ExpAddDropBoxReqStruct

For a list of output fields, see “Output field definitions for ExpDropBoxAddedRspStruct” on page 236.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Unique:** Yes
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**IeMboxIdName**
The Information Exchange logon record ID associated with this dropoff box.

- **Max length:** EXP_MAXLEN_DR_IE_MBOX_ID_NAME (10)
- **Min length:** EXP_MINLEN_DR_IE_MBOX_ID_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the logon record ID must exist in the database.
**EmailOpt**
The option to receive e-mail when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named EMAIL. The filenames are email, email.002, email.003, email.004, and so on.

- **Max length:** EXP_MAXLEN_DR_EMAIL_OPT (1)
- **Min length:** EXP_MINLEN_DR_EMAIL_OPT (1)
- **Default value:** EXP_DFLT_DR_EMAIL_OPT (N)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **N** Do not receive e-mail automatically.
  - **Y** Receive e-mail automatically.

**SysmsgOpt**
The option to receive system messages when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named SYSMSG. The filenames are sysmsg, sysmsg.002, sysmsg.003, sysmsg.004, and so on.

- **Max length:** EXP_MAXLEN_DR_SYSMSG_OPT (1)
- **Min length:** EXP_MINLEN_DR_SYSMSG_OPT (1)
- **Default value:** EXP_DFLT_DR_SYSMSG_OPT (N)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **N** Do not receive system messages automatically.
  - **Y** Receive system messages automatically.
OrdersOpt
This option indicates whether orders are removed from a dropoff box after processing.

- **Max length:** EXP_MAXLEN_DR_ORDERS_OPT (1)
- **Min length:** EXP_MINLEN_DR_ORDERS_OPT (1)
- **Default value:** RM_DFLT_DR_ORDERS_OPT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:**
  - The valid values are:
    - N  Do not remove orders from the dropoff box.
    - Y  Remove the orders from the dropoff box after processing.

AppStatus
This option indicates whether Expedite for Windows should show the status window during the session.

- **Max length:** EXP_MAXLEN_DR_APP_STATUS (1)
- **Min length:** EXP_MINLEN_DR_APP_STATUS (1)
- **Default value:** RM_DFLT_DR_APP_STATUS (Y)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:**
  - The valid values are:
    - N  Do not display the status window.
    - Y  Display the status window.

Reserved1
This is a reserved field; leave it blank.
Length = 1.

Reserved2
This is a reserved field; leave it blank.
Length = 1.
**DropRecovery**
This indicates how a dropoff box is recovered when an error occurs.

Max length: EXP_MAXLEN_DR_DROP_RECOVERY (1)
Min length: EXP_MINLEN_DR_DROP_RECOVERY (1)
Default value: RM_DFLT_DR_DROP_RECOVERY (L)
Required: No
Case sensitive: Yes
Validation: The valid values are:
- C Cancel dropoff box processing if there is an error. If you specify Y for the remove orders option, all orders are removed from the dropoff box. Some orders may process if you do not specify session-level recovery.
- L Leave the dropoff box as is. It is considered disabled by the system. No other dropoff boxes with the same account ID and user ID are processed until this box is processed successfully or is recovered.
- R Reset the dropoff box. Orders that processed successfully are removed from the dropoff box. You must reschedule or manually restart the dropoff box processing later.

**SessRecovery**
The recovery level you want for the Information Exchange session.

Max length: EXP_MAXLEN_DR_SESS_RECOVERY (1)
Min length: EXP_MINLEN_DR_SESS_RECOVERY (1)
Default value: RM_DFLT_DR_SESS_RECOVERY (C)
Required: No
Case sensitive: Yes
Validation: The valid values are:
- C Use checkpoint-level recovery.
- F Use file-level recovery.
- S Use session-level recovery.
Commitdata
The number of bytes to send between checkpoints. This parameter applies only to
checkpoint-level recovery. The value in COMMITDATA must be as large as the value
you specified in MSGSIZE.

- **Max length:** EXP_MAXLEN_DR_COMMITDATA (7)
- **Min length:** EXP_MINLEN_DR_COMMITDATA (4)
- **Max value:** EXP_MAXVAL_DR_COMMITDATA (9999999)
- **Min value:** EXP_MINVAL_DR_COMMITDATA (1000)
- **Default value:** RM_DFLT_DR_COMMITDATA (141000)
- **Required:** No
- **Validation:** Yes, as numeric data between 1000 and 9999999.

Maxmsgs
The maximum number of message segments the program can receive between Infor-
mation Exchange commits.

- **Max length:** EXP_MAXLEN_DR_MAXMSG (4)
- **Min length:** EXP_MINLEN_DR_MAXMSG (1)
- **Max value:** EXP_MAXVAL_DR_MAXMSG (999)
- **Min value:** EXP_MINVAL_DR_MAXMSG (1)
- **Default value:** RM_DFLT_DR_MAXMSG (10)
- **Required:** No
- **Validation:** Yes, as numeric data.

Msgsize
The size of the data segments to send. The value in MSGSIZE must be less than or
equal to the value specified in COMMITDATA.

- **Max length:** EXP_MAXLEN_DR_MSGSIZE (5)
- **Min length:** EXP_MINLEN_DR_MSGSIZE (4)
- **Max value:** EXP_MAXVAL_DR_MSGSIZE (99999)
- **Min value:** EXP_MINVAL_DR_MSGSIZE (1000)
- **Default value:** RM_DFLT_DR_MSGSIZE (47000)
- **Required:** No
- **Validation:** Yes, as numeric data between 1000 and 99999.
XlateTbl
The translation table name for a dropoff box order.

Max length: EXP_MAXLEN_DR_XLATE_TBL (8)
Min length: EXP_MINLEN_DR_XLATE_TBL (1)
Default value: RM_DFLT_DR_XLATE_TBL (IESTDTBL)
Required: No
Case sensitive: No
Validation: Yes, for the translation table file name without the XLT extension.

Overwrite
The action to take when receiving a file that already exists.

Max length: EXP_MAXLEN_DR_OVERWRITE (1)
Min length: EXP_MINLEN_DR_OVERWRITE (1)
Default value: RM_DFLT_DR_OVERWRITE (A)
Required: No
Case sensitive: Yes
Validation: The valid values are:

A  Append the new data to the existing file.
G  Generate a new file name using the file mask specified for the new data.
O  Overwrite the existing file.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddDropBoxReqStruct

**TimeZone**

The time zone to use in the session start command to identify your time zone to Information Exchange. The dates you receive from Information Exchange reflect this time zone.

- **Max length:** EXP_MAXLEN_DR_TIME_ZONE (6)
- **Min length:** EXP_MINLEN_DR_TIME_ZONE (1)
- **Default value:** E0000 (Greenwich)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - E1200 New Zealand
  - E1100 Caroline Islands
  - E1000 Marian Islands
  - E1000D Eastern Australia Daylight
  - E0900 Japan
  - E0800 China
  - E0700 Java
  - E0600 Kazakh
  - E0500 Pakistan
  - E0400 Caspian Sea
  - E0300 Ukraine
  - E0200 Nile
  - E0200D Western Europe Daylight
  - E0100 Europe
  - E0100D British Summer Time
  - E0000 Greenwich
  - W0100 Azores
  - W0200 Oscar
  - W0300 Greenland
  - W0400 Atlantic
  - W0400D U.S. Eastern Daylight
  - W0500 U.S. Eastern
  - W0500D U.S. Central Daylight
  - W0600 U.S. Central
  - W0600D U.S. Mountain Daylight
  - W0700 U.S. Mountain
  - W0800D U.S. Pacific Daylight
  - W0800 U.S. Pacific
  - W0800 Yukon Daylight
  - W0900 Yukon
  - W1000 Hawaii
  - W1100 Bering
Output arguments for ExpAddDropBox

**struct ExpDropBoxAddedRspStruct**

{
    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The status of the dropoff box */
    char Status[EXP_MAXLEN_DR_STATUS */

    /* The Information Exchange logon record ID for this dropoff box */
    char IeMboxIdName[EXP_MAXLEN_DR_IE_MBOX_NAME];

    /* The option to receive data as e-mail */
    char EmailOpt[EXP_MAXLEN_DR_EMAIL_OPT];

    /* The option to receive system messages */
    char SysmsgOpt[EXP_MAXLEN_DR_SYSMSG_OPT];

    /* The option to remove orders from the dropoff box */
    char OrdersOpt[EXP_MAXLEN_DR_ORDERS_OPT];

    /* The option to display session status */
    char AppStatus[EXP_MAXLEN_DR_APP_STATUS];

    /* Reserved field */
    char Reserved1[1];

    /* Reserved field */
    char Reserved2[1];

    /* The dropoff box recovery type */
    char DropRecovery[EXP_MAXLEN_DR_DROP_RECOVERY];

    /* The level of recovery */
    char SessRecovery[EXP_MAXLEN_DR_SESS_RECOVERY];

    /* The number of bytes to send between checkpoints */
    char Commitdata[EXP_MAXLEN_DR_COMMITDATA];

    /* The maximum number of messages received between commits */
    char Maxmsgs[EXP_MAXLEN_DR_MAXMSGGS];

    /* The size of the data segments to send */
    char Msgsize[EXP_MAXLEN_DR_MGSIZE];
}
/* The translation table name for a dropoff box */
char XlateTbl[EXP_MAXLEN_DR_XLATE_TBL];

/*/ The action to take if a file exists */
char Overwrite[EXP_MAXLEN_DR_OVERWRITE];

/*/ The time zone */
char TimeZone[EXP_MAXLEN_DR_TIME_ZONE];

/*/ The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_DR_IE_MBOX_ID];

/*/ Number of orders assigned */
char NumAssigned[EXP_MAXLEN_DR_NUM_ASSIGNED];
};
Output field definitions for ExpDropBoxAddedRspStruct

The response can include any or all of these fields.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Contents:** As entered by the user.

**Status**
This indicates whether the dropoff box is enabled or disabled. If disabled, it is not processed, even if requested.

- **Max length:** EXP_MAXLEN_DR_STATUS (1)
- **Min length:** EXP_MINLEN_DR_STATUS (1)
- **Contents:** One of these valid values:
  - D The logon ID is disabled.
  - E The logon ID is enabled.

**IeMboxIdName**
The Information Exchange logon record ID.

- **Max length:** EXP_MAXLEN_DR_IE_MBOX_ID_NAME (10)
- **Min length:** EXP_MINLEN_DR_IE_MBOX_ID_NAME (1)
- **Contents:** As entered by the user.

**EmailOpt**
The option to receive e-mail messages when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named EMAIL. The filenames are email, email.002, email.003, email.004, and so on.

- **Max length:** EXP_MAXLEN_DR_EMAIL_OPT (1)
- **Min length:** EXP_MINLEN_DR_EMAIL_OPT (1)
- **Contents:** As entered by the user.
SysmsgOpt
The option to receive system messages when dropoff box processing is complete.

Files are stored in the project data directory in a subdirectory named SYSMSGS. The filenames are sysmsg, sysmsg.002, sysmsg.003, sysmsg.004, and so on.

- Max length: EXP_MAXLEN_DR_SYSMSG_OPT (1)
- Min length: EXP_MINLEN_DR_SYSMSG_OPT (1)
- Contents: As entered by the user.

OrdersOpt
The option to remove orders when dropoff box processing is complete.

- Max length: EXP_MAXLEN_DR_ORDERS_OPT (1)
- Min length: EXP_MINLEN_DR_ORDERS_OPT (1)
- Contents: As entered by the user.

AppStatus
Indicates whether Expedite for Windows should show the status window during the session.

- Max length: EXP_MAXLEN_DR_APP_STATUS (1)
- Min length: EXP_MINLEN_DR_APP_STATUS (1)
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - N  Do not display the status window.
  - Y  Display the status window.

Reserved1
This is a reserved field; leave it blank.
Length = 1.

Reserved2
This a reserved field; leave it blank.
Length = 1.
Output field definitions for ExpDropBoxAddedRspStruct

**DropRecovery**
The dropoff box recovery used for this dropoff box.

- **Max length:** EXP_MAXLEN_DR_DROP_RECOVERY (1)
- **Min length:** EXP_MINLEN_DR_DROP_RECOVERY (1)
- **Default value:** RM_DFLT_DR_DROP_RECOVERY (L)
- **Contents:** One of these valid values:
  - **C** Cancel dropoff box processing if there is an error. If you specify Y for the remove orders option, all orders are removed from the dropoff box. Some orders may process if you do not specify session-level recovery.
  - **L** Leave the dropoff box as is. It is considered disabled by the system. No other dropoff boxes with the same account ID and user ID are processed until this box is processed successfully or is recovered.
  - **R** Reset the dropoff box. Orders that processed successfully are removed from the dropoff box. You must reschedule or manually restart the dropoff box processing later.

**SessRecovery**
The recovery level you want for the Information Exchange session.

- **Max length:** EXP_MAXLEN_DR_SESS_RECOVERY (1)
- **Min length:** EXP_MINLEN_DR_SESS_RECOVERY (1)
- **Default value:** RM_DFLT_DR_SESS_RECOVERY (C)
- **Contents:** One of these valid values:
  - **C** Use checkpoint-level recovery.
  - **F** Use file-level recovery.
  - **S** Use session-level recovery.

**Commitdata**
The number of bytes to send between checkpoints.

- **Max length:** EXP_MAXLEN_DR_COMMITDATA (7)
- **Min length:** EXP_MINLEN_DR_COMMITDATA (4)
- **Max value:** EXP_MAXVAL_DR_COMMITDATA (9999999)
- **Min value:** EXP_MINVAL_DR_COMMITDATA (1000)
- **Default value:** RM_DFLT_DR_COMMITDATA (141000)
- **Contents:** Numeric data between 1000 and 9999999.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpDropBoxAddedRspStruct

Maxmsgs
The maximum number of message segments the program can receive between Information Exchange commits.

Max length: EXP_MAXLEN_DR_MAXMSGS (4)
Min length: EXP_MINLEN_DR_MAXMSGS (1)
Max value: EXP_MAXVAL_DR_MAXMSGS (999)
Min value: EXP_MINVAL_DR_MAXMSGS (1)
Default value: RM_DFLT_DR_MAXMSGS (10)
Contents: Numeric data.

Msgsize
The size of the data segments to send.

Max length: EXP_MAXLEN_DR_MSGSIZE (5)
Min length: EXP_MINLEN_DR_MSGSIZE (4)
Max value: EXP_MAXVAL_DR_MSGSIZE (99999)
Min value: EXP_MINVAL_DR_MSGSIZE (1000)
Default value: RM_DFLT_DR_MSGSIZE (47000)
Contents: Numeric data between 1000 and 99999.

XlateTbl
The translation table name for a dropoff box order.

Max length: EXP_MAXLEN_DR_XLATE_TBL (8)
Min length: EXP_MINLEN_DR_XLATE_TBL (1)
Default value: RM_DFLT_DR_XLATE_TBL (IESTDTBL)
Contents: As entered by the user.

Overwrite
The action to take if a file already exists when receiving data.

Max length: EXP_MAXLEN_DR_OVERWRITE (1)
Min length: EXP_MINLEN_DR_OVERWRITE (1)
Default value: RM_DFLT_DR_OVERWRITE (A)
Contents: One of these valid values:
    A  Append the new data to the existing file.
    G  Generate a new file name using the file mask specified for the new data.
    O  Overwrite the existing file.
**TimeZone**

The time zone to use in the session start command to identify your time zone to Information Exchange. The dates you receive from Information Exchange reflect this time zone.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_DR_TIME_ZONE (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_DR_TIME_ZONE (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>E0000 (Greenwich)</td>
</tr>
<tr>
<td>Contents:</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td></td>
<td>E1200 New Zealand</td>
</tr>
<tr>
<td></td>
<td>E1100 Caroline Islands</td>
</tr>
<tr>
<td></td>
<td>E1000 Marian Islands</td>
</tr>
<tr>
<td></td>
<td>E1000D Eastern Australia Daylight</td>
</tr>
<tr>
<td></td>
<td>E0900 Japan</td>
</tr>
<tr>
<td></td>
<td>E0800 China</td>
</tr>
<tr>
<td></td>
<td>E0700 Java</td>
</tr>
<tr>
<td></td>
<td>E0600 Kazakh</td>
</tr>
<tr>
<td></td>
<td>E0500 Pakistan</td>
</tr>
<tr>
<td></td>
<td>E0400 Caspian Sea</td>
</tr>
<tr>
<td></td>
<td>E0300 Ukraine</td>
</tr>
<tr>
<td></td>
<td>E0200 Nile</td>
</tr>
<tr>
<td></td>
<td>E0200D Western Europe Daylight</td>
</tr>
<tr>
<td></td>
<td>E0100 Europe</td>
</tr>
<tr>
<td></td>
<td>E0100D British Summer Time</td>
</tr>
<tr>
<td></td>
<td>E0000 Greenwich</td>
</tr>
<tr>
<td></td>
<td>W0100 Azores</td>
</tr>
<tr>
<td></td>
<td>W0200 Oscar</td>
</tr>
<tr>
<td></td>
<td>W0300 Greenland</td>
</tr>
<tr>
<td></td>
<td>W0400 Atlantic</td>
</tr>
<tr>
<td></td>
<td>W0400D U.S. Eastern Daylight</td>
</tr>
<tr>
<td></td>
<td>W0500 U.S. Eastern</td>
</tr>
<tr>
<td></td>
<td>W0500D U.S. Central Daylight</td>
</tr>
<tr>
<td></td>
<td>W0600 U.S. Central</td>
</tr>
<tr>
<td></td>
<td>W0600D U.S. Mountain Daylight</td>
</tr>
<tr>
<td></td>
<td>W0700 U.S. Mountain</td>
</tr>
<tr>
<td></td>
<td>W0800D U.S. Pacific Daylight</td>
</tr>
<tr>
<td></td>
<td>W0800 U.S. Pacific</td>
</tr>
<tr>
<td></td>
<td>W0800 Yukon Daylight</td>
</tr>
<tr>
<td></td>
<td>W0900 Yukon</td>
</tr>
<tr>
<td></td>
<td>W1000 Hawaii</td>
</tr>
<tr>
<td></td>
<td>W1100 Bering</td>
</tr>
</tbody>
</table>
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpDropBoxAddedRspStruct

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** `EXP_MAXLEN_DR_IE_MBOX_ID` (16)
- **Min length:** `EXP_MINLEN_DR_IE_MBOX_ID` (1)
- **Contents:** As entered by the user.

**NumAssigned**
This indicates the number of orders assigned to the dropoff box.

- **Max length:** `EXP_MAXLEN_DR_NUM_ASSIGNED` (5)
- **Min length:** `EXP_MINLEN_DR_NUM_ASSIGNED` (1)
- **Contents:** Numeric data.
ExpDeleteDropBox

This function deletes the dropoff box from the MailBox database. If the dropoff box name specified does not exist, the return code is set to 4.

If the dropoff box has any receive orders assigned to it, they are unassigned before the dropoff box is deleted. Likewise, if the dropoff box has any send orders assigned to it, they are unassigned before the dropoff box is deleted.

Function prototype for ExpDeleteDropBox

```c
int ExpDeleteDropBox(
    struct ExpDeleteDropBoxReqStruct * Input
);
```

Input arguments for ExpDeleteDropBox

```c
struct ExpDeleteDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DRBX_NAME];
};
```
Input field definitions for ExpDeleteDropBoxReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the dropoff box must exist in the database.
ExpListDropBox

This function lists the dropoff boxes in the Mailbox database that correspond to the query formed in the request structure. If the query results in no dropoff boxes found, the return code is set to 4.

Function prototype for ExpListDropBox

```c
int ExpListDropBox(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpDropBoxListedRspStruct *Output
);
```

Input arguments for ExpListDropBox

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The specific parameter ID to search on */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The specific parameter ID to sort on */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to receive the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListDropBoxReqStruct

For a list of the output fields, see “Output field definitions for ExpDropBoxListedRspStruct” on page 250.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjName</td>
<td>The project name.</td>
<td>EXP_MAXLEN_PJ_PROJ_NAME (20)</td>
<td>EXP_MINLEN_PJ_PROJ_NAME (1)</td>
<td>Yes</td>
<td>No</td>
<td>Yes, the project must be open.</td>
</tr>
</tbody>
</table>

**SearchParm**
The field on which to search.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchParm</td>
<td>The field on which to search.</td>
<td>EXP_MAXLEN_CF_SEARCH_PARM (15)</td>
<td>EXP_MINLEN_SEARCH_PARM (1)</td>
<td>Yes, if a SearchValue is specified.</td>
<td>Yes</td>
<td>Yes, the search field is Name.</td>
</tr>
</tbody>
</table>

**SearchOp**
The search operator to use.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchOp</td>
<td>The search operator to use.</td>
<td>EXP_MAXLEN_CF_SEARCH_OPER (2)</td>
<td>EXP_MINLEN_SEARCH_OPER (1)</td>
<td>Yes, if a SearchParm is specified.</td>
<td>No</td>
<td>Yes, the valid characters are: =, &lt;, &gt;, &lt;=, or &gt;=.</td>
</tr>
</tbody>
</table>
**SearchValue**  
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)  
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)  
- **Required:** Yes, if a SearchParm is specified.  
- **Case sensitive:** Yes  
- **Validation:** No

**SortParm**  
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)  
- **Min length:** EXP_MINLEN_SORT_PARM (1)  
- **Required:** Yes, if a SortOrder is specified.  
- **Case sensitive:** Yes  
- **Validation:** Yes, The sort field is Name.

**SortOrder**  
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)  
- **Min length:** EXP_MINLEN_SORT_ORDER (1)  
- **Required:** No  
- **Case sensitive:** Yes  
- **Validation:** The valid values are:
  
  RM_SORT_ASCENDING (A)  
  Sort the list in ascending order.
  
  RM_SORT_DESCENDING (D)  
  Sort the list in descending order.
Input field definitions for ExpListDropBoxReqStruct

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListDropBox

```c
struct ExpDropBoxListedRspStruct {
    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The dropoff box error status */
    char Status[EXP_MAXLEN_DR_STATUS];

    /* The Information Exchange logon record ID for this dropoff box */
    char IeMboxIdName[EXP_MAXLEN_DR_IE_MBOX_ID_NAME];

    /* The option to receive data as e-mail */
    char EmailOpt[EXP_MAXLEN_DR_EMAIL_OPT];

    /* The option to receive system messages */
    char SysmsgsOpt[EXP_MAXLEN_DR_SYSMSGS_OPT];

    /* The option to remove orders from a dropoff box */
    char OrdersOpt[EXP_MAXLEN_DR_ORDERS_OPT];

    /* The option to display session status */
    char AppStatus[EXP_MAXLEN_DR_APP_STATUS];

    /* Reserved field */
    char Reserved2[1];

    /* Reserved field */
    char Reserved2[1];

    /* The dropoff recovery type */
    char DropRecovery[EXP_MAXLEN_DR_DROP_RECOVERY];

    /* The recovery level */
    char SessRecovery[EXP_MAXLEN_DR_SESS_RECOVERY];

    /* The number of bytes to send between checkpoints */
    char Commitdata[EXP_MAXLEN_DR_COMMITDATA];

    /* The maximum messages received between commits */
    char Maxmsgs[EXP_MAXLEN_DR_MAXMSGS];

    /* The data segments for sending */
    char Msgsize[EXP_MAXLEN_DR_MSGSIZE];

    /* The translation table name for a dropoff box */
    char XlateTbl[EXP_MAXLEN_DR_XLATE_TBL];
};
```
Output arguments for ExpListDropBox

/* The action to take if a file exists */
char Overwrite[EXP_MAXLEN_DR_OVERWRITE];

/* The time zone */
char TimeZone[EXP_MAXLEN_DR_TIME_ZONE];

/* The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_DR_IE_MBOX_ID];

/* The number of orders assigned */
char NumAssigned[EXP_MAXLEN_DR_NUM_ASSIGNED];

};
Output field definitions for ExpDropBoxListedRspStruct

The response can include any or all of these fields.

**Entries Returned**
The number of dropoff boxes copied into the output structure.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Contents:** As entered by the user.

**Status**
The dropoff box is enabled or disabled. If it is disabled, it is not available for processing even if processing is scheduled or requested.

- **Max length:** EXP_MAXLEN_DR_STATUS (1)
- **Min length:** EXP_MINLEN_DR_STATUS (1)
- **Contents:** One of these valid values:
  - D The dropoff box is disabled.
  - E The dropoff box is enabled.

**IeMboxIdName**
The Information Exchange logon record ID.

- **Max length:** EXP_MAXLEN_DR_IE_MBOX_ID_NAME (10)
- **Min length:** EXP_MINLEN_DR_IE_MBOX_ID_NAME (1)
- **Contents:** Alphanumeric data.
EmailOpt
The option to receive e-mail messages (files with the user message class of FFMSG001) when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named EMAIL. The filenames are email, email.002, email.003, email.004, and so on.

- **Max length**: EXP_MAXLEN_DR_EMAIL_OPT (1)
- **Min length**: EXP_MINLEN_DR_EMAIL_OPT (1)
- **Required**: No
- **Case sensitive**: No
- **Validation**: The valid values are:
  - N  Do not receive e-mail automatically.
  - Y  Receive e-mail automatically.

SysmsgOpt
This indicates whether system messages (files received from account *SYSTEM* and user ID *ERRMSG*) were received when dropoff box processing completed. Files are stored in the project data directory in a subdirectory named SYSMSGS. The filenames are sysmsg, sysmsg.002, sysmsg.003, sysmsg.004, and so on.

- **Max length**: EXP_MAXLEN_DR_SYSMSG_OPT (1)
- **Min length**: EXP_MINLEN_DR_SYSMSG_OPT (1)
- **Contents**: One of these valid values:
  - N  Do not receive system messages automatically.
  - Y  Receive system messages automatically.

OrdersOpt
This option indicates whether orders are removed from a dropoff box after processing.

- **Max length**: EXP_MAXLEN_DR_ORDERS_OPT (1)
- **Min length**: EXP_MINLEN_DR_ORDERS_OPT (1)
- **Contents**: One of these valid values:
  - N  Do not remove orders from the dropoff box.
  - Y  Remove the orders from the dropoff box after processing.
**AppStatus**
This indicates whether Expedite for Windows should show the status window during the session.

- **Max length:** EXP_MAXLEN_DR_APP_STATUS (1)
- **Min length:** EXP_MINLEN_DR_APP_STATUS (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Do not display the status window.
  - Y  Display the status window.

**Reserved1**
This is a reserved field; leave it blank.
Length = 1.

**Reserved2**
This is a reserved field; leave it blank.
Length = 1.

**DropRecovery**
The dropoff box recovery level used for this dropoff box.

- **Max length:** EXP_MAXLEN_DR_DROP_RECOVERY (1)
- **Min length:** EXP_MINLEN_DR_DROP_RECOVERY (1)
- **Contents:** One of these valid values:
  - C  Cancel dropoff box processing if there is an error. If you specified Y for the remove orders option, all orders will be removed from the dropoff box. Some orders may have processed if you did not specify session-level recovery.
  - L  Leave the dropoff box as is. It is considered disabled by the system. No other dropoff boxes with the same account and user ID are processed until this box is successfully processed or recovered.
  - R  Reset the dropoff box. Orders that have successfully processed are removed from the dropoff box. You must reschedule or manually restart the dropoff box processing for a later time.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpDropBoxListedRspStruct

**SessRecovery**
The recovery level for the Information Exchange session.

- **Max length:** EXP_MAXLEN_DR_SESS_RECOVERY (1)
- **Min length:** EXP_MINLEN_DR_SESS_RECOVERY (1)
- **Contents:** One of these valid values:
  - C Checkpoint-level recovery.
  - F File-level recovery.
  - S Session-level recovery.

**Commitdata**
The number of bytes to send between checkpoints. This parameter applies only to checkpoint-level recovery. COMMITDATA must be as large as the value you specified in MSGSIZE.

- **Max length:** EXP_MAXLEN_DR_COMMITDATA (7)
- **Min length:** EXP_MINLEN_DR_COMMITDATA (4)
- **Max value:** EXP_MAXVAL_DR_COMMITDATA (9999999)
- **Min value:** EXP_MINVAL_DR_COMMITDATA (141000)
- **Contents:** Numeric data between 1000 and 9999999.

**Maxmsgs**
The maximum number of message segments the program can receive between Information Exchange commits.

- **Max length:** EXP_MAXLEN_DR_MAXMSGS (4)
- **Min length:** EXP_MINLEN_DR_MAXMSGS (1)
- **Max value:** EXP_MAXVAL_DR_MAXMSGS (999)
- **Min value:** EXP_MINVAL_DR_MAXMSGS (1)
- **Contents:** Numeric data.

**Msgsize**
The size of the data segments to send. The value of MSGSIZE must be less than or equal to the value you specified in COMMITDATA.

- **Max length:** EXP_MAXLEN_DR_MSGSIZE (5)
- **Min length:** EXP_MINLEN_DR_MSGSIZE (4)
- **Max value:** EXP_MAXVAL_DR_MSGSIZE (999999)
- **Min value:** EXP_MINVAL_DR_MSGSIZE (1000)
- **Contents:** Numeric data between 1000 and 999999.
**XlateTbl**
The translation table name for a dropoff box order.

- Max length: EXP_MAXLEN_DR_XLATE_TBL (8)
- Min length: EXP_MINLEN_DR_XLATE_TBL (1)
- Contents: Alphanumeric data.

**Overwrite**
The action to take when receiving a file that already exists.

- Max length: EXP_MAXLEN_DR_OVERWRITE (1)
- Min length: EXP_MINLEN_DR_OVERWRITE (1)
- Contents: One of these valid values:
  - A  Append the new data to the existing file.
  - G  Generate a new file name using the file mask specified for the new data.
  - O  Overwrite the existing file.
TimeZone
The time zone used on the session start command to identify your time zone to Information Exchange. The dates you receive in responses from Information Exchange will reflect this time zone.

Max length: EXP_MAXLEN_DR_TIME_ZONE (6)
Min length: EXP_MINLEN_DR_TIME_ZONE (1)
Contents: One of these valid values:

- E1200 New Zealand
- E1100 Caroline Islands
- E1000 Marian Islands
- E1000D Eastern Australia Daylight
- E0900 Japan
- E0800 China
- E0700 Java
- E0600 Kazakh
- E0500 Pakistan
- E0400 Caspian Sea
- E0300 Ukraine
- E0200 Nile
- E0200D Western Europe Daylight
- E0100 Europe
- E0100D British Summer Time
- E0000 Greenwich
- W0100 Azores
- W0200 Oscar
- W0300 Greenland
- W0400 Atlantic
- W0400D U.S. Eastern Daylight
- W0500 U.S. Eastern
- W0500D U.S. Central Daylight
- W0600 U.S. Central
- W0600D U.S. Mountain Daylight
- W0700 U.S. Mountain
- W0800D U.S. Pacific Daylight
- W0800 U.S. Pacific
- W0800 Yukon Daylight
- W0900 Yukon
- W1000 Hawaii
- W1100 Bering
**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- Max length: EXP_MAXLEN_DR_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_DR_IE_MBOX_ID (9)
- Contents: Alphanumeric data.

**NumAssigned**
This indicates the number of orders assigned to the dropoff box.

- Max length: EXP_MAXLEN_DR_NUM_ASSIGNED (5)
- Min length: EXP_MINLEN_DR_NUM_ASSIGNED (1)
- Contents: Numeric data.
ExpReplaceDropBox

This function replaces information about the dropoff box. If the dropoff box name specified does not exist, the return code is set to 8.

Function prototype for ExpReplaceDropBox

```c
int ExpReplaceDropBox(
    struct ExpReplaceDropBoxReqStruct * Input,
    struct ExpDropBoxReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceDropBox

```c
struct ExpReplaceDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The Information Exchange logon record ID for this dropoff box */
    char IeMboxIdName[EXP_MAXLEN_DR_IE_MBOX_ID_NAME];

    /* The option to receive e-mail at end of session */
    char EmailOpt[EXP_MAXLEN_DR_EMAIL_OPT];

    /* The option to receive system messages at end of session */
    char SysmsgOpt[EXP_MAXLEN_DR_SYSMSG_OPT];

    /* The option to remove orders from the dropoff box */
    char OrdersOpt[EXP_MAXLEN_DR_ORDERS_OPT];

    /* The option to display session status */
    char AppStatus[EXP_MAXLEN_DR_APP_STATUS];

    /* Reserved field */
    char Reserved2[1];

    /* Reserved field */
    char Reserved1[1];

    /* The dropoff recovery type */
    char DropRecovery[EXP_MAXLEN_DR_DROP_RECOVERY];
}
```
/* The recovery level */
char SessRecovery[EXP_MAXLEN_DR_SESS_RECOVERY];

/* The number of bytes to send between checkpoints */
char Commitdata[EXP_MAXLEN_DR_COMMITDATA];

/* The maximum number of messages received between commits */
char Maxmsgs[EXP_MAXLEN_DR_MAXMSGS];

/* The size of the data segments to send */
char Msgsize[EXP_MAXLEN_DR_MSGSIZE];

/* The translation table name for a dropoff box */
char XlateTbl[EXP_MAXLEN_DR_XLATE_TBL];

/* The action to take if a file exists */
char Overwrite[EXP_MAXLEN_DR_OVERWRITE];

/* The time zone */
char TimeZone[EXP_MAXLEN_DR_TIME_ZONE];

};
Input field definitions for ExpReplaceDropBoxReqStruct

For a list of the output fields, see “Output field definitions for ExpDropBoxReplacedRspStruct” on page 268.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the dropoff box name must exist in the database.

**IeMboxIdName**
The Information Exchange logon record ID associated with this dropoff box.

- **Max length:** EXP_MAXLEN_DR_IE_MBOX_ID_NAME (10)
- **Min length:** EXP_MINLEN_DR_IE_MBOX_ID_NAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, the logon record must exist in the database.
**EmailOpt**
The option to receive e-mail messages (files with user message class FFMSG001) when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named EMAIL. The filenames are email, email.002, email.003, email.004, and so on.

Max length: EXP_MAXLEN_DR_EMAIL_OPT (1)
Min length: EXP_MINLEN_DR_EMAIL_OPT (1)
Default value: EXP_DFLT_DR_EMAIL_OPT (N)
Required: No
Case sensitive: No
Validation: The valid values are:
   - N  Do not receive e-mail automatically.
   - Y  Receive e-mail automatically.

**SysmsgOpt**
The option to receive system messages (files from account ID *SYSTEM* and user ID *ERRMSG*) are received when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named SYSMSGS. The filenames are sysmsg, sysmsg.002, sysmsg.003, sysmsg.004, and so on.

Max length: EXP_MAXLEN_DR_SYSMSG_OPT (1)
Min length: EXP_MINLEN_DR_SYSMSG_OPT (1)
Default value: EXP_DFLT_DR_SYSMSG_OPT (N)
Required: No
Case sensitive: No
Validation: The valid values are:
   - N  Do not receive system messages automatically.
   - Y  Receive system messages automatically.
OrdersOpt
The option to remove orders from a dropoff box after processing.

Max length: EXP_MAXLEN_DR_ORDERS_OPT (1)
Min length: EXP_MINLEN_DR_ORDERS_OPT (1)
Default value: RM_DFLT_DR_ORDERS_OPT (N)
Required: No
Case sensitive: No
Validation: The valid values are:
N  Do not remove orders from the dropoff box.
Y  Remove orders from the dropoff box after processing.

AppStatus
This indicates whether Expedite for Windows should show the status window during the session.

Max length: EXP_MAXLEN_DR_APP_STATUS (1)
Min length: EXP_MINLEN_DR_APP_STATUS (1)
Default value: RM_DFLT_DR_APP_STATUS (Y)
Required: No
Case sensitive: Yes
Validation: The valid values are:
N  Do not display the status window.
Y  Display the status window.

Reserved1
This is a reserved field; leave it blank.
Length = 1.

Reserved2
This is a reserved field; leave it blank.
Length = 1.
### DropRecovery

This indicates how a dropoff box is recovered when an error occurs.

| Max length: | EXP_MAXLEN_DR_DROP_RECOVERY (1) |
| Min length: | EXP_MINLEN_DR_DROP_RECOVERY (1) |
| Default value: | RM_DFLT_DR_DROP_RECOVERY (L) |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
| C | Cancel dropoff box processing if there is an error. If you specify Y for the remove orders option, all orders are removed from the dropoff box. Some orders may process if you do not specify session-level recovery. |
| L | Leave the dropoff box as is. It is considered disabled by the system. No other dropoff boxes with the same account ID and user ID are processed until this box processes successfully or is recovered. |
| R | Reset the dropoff box. Orders that process successfully are removed from the dropoff box. You must reschedule or manually restart the dropoff box processing later. |

### SessRecovery

The recovery level you want for the Information Exchange session.

| Max length: | EXP_MAXLEN_DR_SESS_RECOVERY (1) |
| Min length: | EXP_MINLEN_DR_SESS_RECOVERY (1) |
| Default value: | RM_DFLT_DR_SESS_RECOVERY (C) |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
| C | Use checkpoint-level recovery. |
| F | Use file-level recovery. |
| S | Use session-level recovery. |
Commitdata
The number of bytes to send between checkpoints. This applies only to checkpoint-level recovery. The value in COMMITDATA must be as large as the value you specified in MSGSIZE.

- Max length: EXP_MAXLEN_DR_COMMITDATA (7)
- Min length: EXP_MINLEN_DR_COMMITDATA (4)
- Max value: EXP_MAXVAL_DR_COMMITDATA (9999999)
- Min value: EXP_MINVAL_DR_COMMITDATA (1000)
- Default value: RM_DFLT_DR_COMMITDATA (141000)
- Required: No
- Validation: Yes, for numeric data between 1000 and 9999999.

Maxmsgs
The maximum number of message segments the program can receive between Information Exchange commits.

- Max length: EXP_MAXLEN_DR_MAXMSGS (4)
- Min length: EXP_MINLEN_DR_MAXMSGS (1)
- Max value: EXP_MAXVAL_DR_MAXMSGS (999)
- Min value: EXP_MINVAL_DR_MAXMSGS (1)
- Default value: RM_DFLT_DR_MAXMSGS (10)
- Required: No
- Validation: Yes, for numeric data.

Msgsize
The size of the data segments to send. The value in MSGSIZE must be less than or equal to the value you specified in COMMITDATA.

- Max length: EXP_MAXLEN_DR_MSGSIZE (5)
- Min length: EXP_MINLEN_DR_MSGSIZE (4)
- Max value: EXP_MAXVAL_DR_MSGSIZE (99999)
- Min value: EXP_MINVAL_DR_MSGSIZE (1000)
- Default value: RM_DFLT_DR_MSGSIZE (47000)
- Required: No
- Validation: Yes, for numeric data between 1000 and 99999.
XlateTbl
The translation table name for a drop-off box order.

Max length: \( \text{EXP\_MAXLEN\_DR\_XLATE\_TBL} \) (8)
Min length: \( \text{EXP\_MINLEN\_DR\_XLATE\_TBL} \) (1)
Default value: \( \text{RM\_DFLT\_DR\_XLATE\_TBL} \) (IESTDTBL)
Required: No
Case sensitive: No
Validation: Yes, it must contain the file name for the translation table without the XLT extension.

Overwrite
The action to take if a file already exists when receiving data.

Max length: \( \text{EXP\_MAXLEN\_DR\_OVERWRITE} \) (1)
Min length: \( \text{EXP\_MINLEN\_DR\_OVERWRITE} \) (1)
Default value: \( \text{RM\_DFLT\_DR\_OVERWRITE} \) (A)
Required: No
Case sensitive: Yes
Validation: The valid values are:
\( A \) Append the new data to the existing file.
\( G \) Generate a new file name using the file mask specified for the new data.
\( O \) Overwrite the existing file.
**TimeZone**

The time zone to use in the session start command to identify your time zone to Information Exchange. The dates you receive from Information Exchange reflect this time zone.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_DR_TIME_ZONE (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_DR_TIME_ZONE (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>E0000 (Greenwich)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td>E1200  New Zealand</td>
</tr>
<tr>
<td></td>
<td>E1100  Caroline Islands</td>
</tr>
<tr>
<td></td>
<td>E1000  Marian Islands</td>
</tr>
<tr>
<td></td>
<td>E1000D  Eastern Australia Daylight</td>
</tr>
<tr>
<td></td>
<td>E0900  Japan</td>
</tr>
<tr>
<td></td>
<td>E0800  China</td>
</tr>
<tr>
<td></td>
<td>E0700  Java</td>
</tr>
<tr>
<td></td>
<td>E0600  Kazakh</td>
</tr>
<tr>
<td></td>
<td>E0500  Pakistan</td>
</tr>
<tr>
<td></td>
<td>E0400  Caspian Sea</td>
</tr>
<tr>
<td></td>
<td>E0300  Ukraine</td>
</tr>
<tr>
<td></td>
<td>E0200  Nile</td>
</tr>
<tr>
<td></td>
<td>E0200D  Western Europe Daylight</td>
</tr>
<tr>
<td></td>
<td>E0100  Europe</td>
</tr>
<tr>
<td></td>
<td>E0100D  British Summer Time</td>
</tr>
<tr>
<td></td>
<td>E0000  Greenwich</td>
</tr>
<tr>
<td></td>
<td>W0100  Azores</td>
</tr>
<tr>
<td></td>
<td>W0200  Oscar</td>
</tr>
<tr>
<td></td>
<td>W0300  Greenland</td>
</tr>
<tr>
<td></td>
<td>W0400  Atlantic</td>
</tr>
<tr>
<td></td>
<td>W0400D  U.S. Eastern Daylight</td>
</tr>
<tr>
<td></td>
<td>W0500  U.S. Eastern</td>
</tr>
<tr>
<td></td>
<td>W0500D  U.S. Central Daylight</td>
</tr>
<tr>
<td></td>
<td>W0600  U.S. Central</td>
</tr>
<tr>
<td></td>
<td>W0600D  U.S. Mountain Daylight</td>
</tr>
<tr>
<td></td>
<td>W0700  U.S. Mountain</td>
</tr>
<tr>
<td></td>
<td>W0800D  U.S. Pacific Daylight</td>
</tr>
<tr>
<td></td>
<td>W0800  U.S. Pacific</td>
</tr>
<tr>
<td></td>
<td>W0800  Yukon</td>
</tr>
<tr>
<td></td>
<td>W1000  Hawaii</td>
</tr>
<tr>
<td></td>
<td>W1100  Bering</td>
</tr>
</tbody>
</table>
Output arguments for ExpReplaceDropBox

struct ExpDropBoxReplacedRspStruct
{
    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The dropoff box error status */
    char Status[EXP_MAXLEN_DR_STATUS *]

    /* The Information Exchange logon record ID for this dropoff box */
    char IeMboxIdName[EXP_MAXLEN_DR_IE_MBOX_NAME];

    /* The option to receive e-mail at end of session */
    char EmailOpt[EXP_MAXLEN_DR_EMAIL_OPT];

    /* The option to receive system messages at end of session */
    char SysmsgOpt[EXP_MAXLEN_DR_SYSMSG_OPT];

    /* The option to remove orders from the dropoff box */
    char OrdersOpt[EXP_MAXLEN_DR_ORDERS_OPT];

    /* The option to display session status */
    char AppStatus[EXP_MAXLEN_DR_APP_STATUS];

    /* Reserved field */
    char Reserved1[1];

    /* Reserved field */
    char Reserved2[1];

    /* The dropoff box recovery type */
    char DropRecovery[EXP_MAXLEN_DR_DROP_RECOVERY];

    /* The level of recovery */
    char SessRecovery[EXP_MAXLEN_DR_SESS_RECOVERY];

    /* The number of bytes to send between checkpoints */
    char Commitdata[EXP_MAXLEN_DR_COMMITDATA];

    /* The maximum number of messages received between commits */
    char Maxmsgs[EXP_MAXLEN_DR_MAXMSGS];

    /* The size of the data segments to send */
    char Msgsize[EXP_MAXLEN_DR_MSGSIZE];
/* The translation table name for a dropoff box */
char XlateTbl[EXP_MAXLEN_DR_XLATE_TBL];

/* The action to take if a file exists */
char Overwrite[EXP_MAXLEN_DR_OVERWRITE];

/* The time zone */
char TimeZone[EXP_MAXLEN_DR_TIME_ZONE];

/* The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_DR_IE_MBOX_ID];

/* The number of orders assigned */
char NumAssigned[EXP_MAXLEN_DR_NUM_ASSIGNED];

};
Output field definitions for ExpDropBoxReplacedRspStruct

The response can include any or all of these fields.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_DR_DROP_BOX_NAME (1)
- **Contents:** As entered by the user.

**Status**
The dropoff box is enabled or disabled. If disabled, it is not processed even if scheduled or requested.

- **Max length:** EXP_MAXLEN_DR_STATUS (1)
- **Min length:** EXP_MINLEN_DR_STATUS (1)
- **Contents:** One of these valid values:
  - **D** The dropoff box is disabled.
  - **E** The dropoff box is enabled.

**IeMboxIdName**
The Information Exchange logon record ID.

- **Max length:** EXP_MAXLEN_DR_IE_MBOX_ID_NAME (10)
- **Min length:** EXP_MINLEN_DR_IE_MBOX_ID_NAME (1)
- **Contents:** As entered by the user.

**EmailOpt**
The option to receive e-mail messages (files with user message class FFMSG001) when dropoff box processing is complete. Files are stored in the project data directory in a subdirectory named EMAIL. The filenames are email, email.002, email.003, email.004, and so on.

- **Max length:** EXP_MAXLEN_DR_EMAIL_OPT (1)
- **Min length:** EXP_MINLEN_DR_EMAIL_OPT (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **N** Do not receive e-mail automatically.
  - **Y** Receive e-mail automatically.
**SysmsgOpt**
This indicates whether system messages (files from account ID *SYSTEM* and user ID *ERRMSG*) were received when dropoff box processing completed. Files are stored in the project data directory in a subdirectory named SYSMSGS. The filenames are sysmsg, sysmsg.002, sysmsg.003, sysmsg.004, and so on.

- **Max length:** EXP_MAXLEN_DR_SYSMSG_OPT (1)
- **Min length:** EXP_MINLEN_DR_SYSMSG_OPT (1)
- **Contents:** As entered by the user.

**OrdersOpt**
The orders are removed from a dropoff box after processing.

- **Max length:** EXP_MAXLEN_DR_ORDERS_OPT (1)
- **Min length:** EXP_MINLEN_DR_ORDERS_OPT (1)
- **Contents:** As entered by the user.

**AppStatus**
This indicates whether Expedite for Windows should show the status window during the session.

- **Max length:** EXP_MAXLEN_DR_APP_STATUS (1)
- **Min length:** EXP_MINLEN_DR_APP_STATUS (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Do not display the status window.
  - Y  Display the status window.

**Reserved1**
This is a reserved field; leave it blank.
Length = 1.

**Reserved2**
This is a reserved field; leave it blank.
Length = 1.
Drop Recovery
The dropoff box recovery level used for this dropoff box.

Max length: EXP_MAXLEN_DR_DROP_RECOVERY (1)
Min length: EXP_MINLEN_DR_DROP_RECOVERY (1)
Contents: One of these valid values:

C  Cancel dropoff box processing if there is an error. If you specify Y for the remove orders option, all orders are removed from the dropoff box. Some orders may process if you do not specify session-level recovery.

L  Leave the dropoff box as is. It is considered disabled by the system. No other dropoff boxes with the same account ID and user ID are processed until this box processes successfully or is recovered.

R  Reset the dropoff box. Orders that process successfully are removed from the dropoff box. You must reschedule or manually restart the dropoff box processing later.

SessRecovery
The recovery level you want for the Information Exchange session.

Max length: EXP_MAXLEN_DR_SESS_RECOVERY (1)
Min length: EXP_MINLEN_DR_SESS_RECOVERY (1)
Contents: Numeric data.

Commitdata
The number of bytes to send between checkpoints.

Max length: EXP_MAXLEN_DR_COMMITDATA (7)
Min length: EXP_MINLEN_DR_COMMITDATA (4)
Max value: EXP_MAXV AL_DR_COMMITDATA (9999999)
Min value: EXP_MINV AL_DR_COMMITDATA (1000)
Contents: Numeric data between 1000 and 9999999.
Maxmsgs
The maximum number of message segments the program can receive between Information Exchange commits.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Max value</th>
<th>Min value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maxmsgs</td>
<td>The maximum number of message segments</td>
<td>EXP_MAXLEN_DR_MAXMSG (4)</td>
<td>EXP_MINLEN_DR_MAXMSG (1)</td>
<td>EXP_MAXVAL_DR_MAXMSG (999)</td>
<td>EXP_MINVAL_DR_MAXMSG (1)</td>
<td>Numeric data</td>
</tr>
</tbody>
</table>

Msgsize
The size of the data segments to send.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Max value</th>
<th>Min value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msgsize</td>
<td>The size of the data segments to send</td>
<td>EXP_MAXLEN_DR_MSGSIZE (5)</td>
<td>EXP_MINLEN_DR_MSGSIZE (4)</td>
<td>EXP_MAXVAL_DR_MSGSIZE (99999)</td>
<td>EXP_MINVAL_DR_MSGSIZE (1000)</td>
<td>Numeric data between 1000 and 99999</td>
</tr>
</tbody>
</table>

XlateTbl
The translation table name for a dropoff box order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>XlateTbl</td>
<td>The translation table name for a dropoff</td>
<td>EXP_MAXLEN_DR_XLATE_TBL (8)</td>
<td>EXP_MINLEN_DR_XLATE_TBL (1)</td>
<td>As entered by the user.</td>
</tr>
</tbody>
</table>

Overwrite
The action to take if a file already exists when receiving data.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required: Yes</th>
<th>Contents: One of these valid values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwrite</td>
<td>The action to take if a file already</td>
<td>EXP_MAXLEN_DR_OVERWRITE (1)</td>
<td>EXP_MINLEN_DR_OVERWRITE (1)</td>
<td>No</td>
<td>A Append the new data to the existing file.</td>
</tr>
<tr>
<td></td>
<td>exists when receiving data.</td>
<td></td>
<td></td>
<td></td>
<td>G Generate a new file name using the file mask specified for the new data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O Overwrite the existing file.</td>
</tr>
</tbody>
</table>
TimeZone

The time zone to use in the session start command to identify your time zone to Information Exchange. The dates you receive from Information Exchange reflect this time zone.

Max length: EXP_MAXLEN_DR_TIME_ZONE (6)
Min length: EXP_MINLEN_DR_TIME_ZONE (1)
Contents: One of these valid values:

- E1200 New Zealand
- E1100 Caroline Islands
- E1000 Marian Islands
- E1000D Eastern Australia Daylight
- E0900 Japan
- E0800 China
- E0700 Java
- E0600 Kazakh
- E0500 Pakistan
- E0400 Caspian Sea
- E0300 Ukraine
- E0200 Nile
- E0200D Western Europe Daylight
- E0100 Europe
- E0100D British Summer Time
- E0000 Greenwich
- W0100 Azores
- W0200 Oscar
- W0300 Greenland
- W0400 Atlantic
- W0400D U.S. Eastern Daylight
- W0500 U.S. Eastern
- W0500D U.S. Central Daylight
- W0600 U.S. Central
- W0600D U.S. Mountain Daylight
- W0700 U.S. Mountain
- W0800D U.S. Pacific Daylight
- W0800 U.S. Pacific
- W0900 Yukon Daylight
- W1000 Yukon
- W1100 Hawaii
- W1100 Bering
IeMboxId
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next are the user ID.

- Max length: EXP_MAXLEN_DR_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_DR_IE_MBOX_ID (1)
- Contents: Printable data.

NumAssigned
This indicates the number of orders assigned to the dropoff box.

- Max length: EXP_MAXLEN_DR_NUM_ASSIGNED (5)
- Min length: EXP_MINLEN_DR_NUM_ASSIGNED (1)
- Contents: Numeric data.
Receive orders

These are the functions, the input and output structures, and the field descriptions used to work with receive orders in the MailBox database.

ExpAddRecvOrdr

This function adds a new receive order to the MailBox database. If the receive order name already exists, the return code is set to 8.

Function prototype for ExpAddRecvOrdr

```c
int ExpAddRecvOrdr(
    struct ExpAddRecvOrdrReqStruct * Input,
    struct ExpRecvOrdrAddedRspStruct * Output
);
```
Input arguments for ExpAddRecvOrdr

```
struct ExpAddRecvOrdrReqStruct {
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The order name for a receive */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];

    /* The order description */
    char OrderDescr[EXP_MAXLEN_RV_ORDER_DESCR];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_RV_IE_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxID[EXP_MAXLEN_RV_IE_MBOX_ID];

    /* The table type for the alias table */
    char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_RV_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_RV_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_RV_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_RV_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_RV_ADDR_TYPE];

    /* The Information Exchange message class */
    char MsgClass[EXP_MAXLEN_RV_MSG_CLASS];

    /* The file name for the received data */
    char Fileid[EXP_MAXLEN_RV_FILEID];

    /* The archive ID of the received data */
    char Archiveid[EXP_MAXLEN_RV_ARCHIVEID];

    /* The option to receive one file or all files that match */
    char Allfiles[EXP_MAXLEN_RV_ALLFILES];
}
```
/* The multiple files parameter for the receive */
char Multfiles[EXP_MAXLEN_RV_MULTFILES];

/* The receipt of files with the original name */
char Origfile[EXP_MAXLEN_RV_ORIGFILE];

/* The format for the received data */
char Format[EXP_MAXLEN_RV_FORMAT];

/* The EDI record size */
char Recordsize[EXP_MAXLEN_RV_RECORDSIZE];

/* The processing for line length delimiters */
char Processlen[EXP_MAXLEN_RV_PROCESSLEN];

/* The automatic processing for EDI data */
char Autoedi[EXP_MAXLEN_RV_AUTOEDI];

/* The option for EDI data processing */
char Ediopt[EXP_MAXLEN_RV_EDIOPT];

/* Reserved field */
char Reserved1[20];

/* The requeued parameter for a receive */
char Requeued[EXP_MAXLEN_RV_REQUEUED];

/* The controls for the removal of EOF characters */
char Removeeof[EXP_MAXLEN_RV_REMOVEEOF];

/* The option to receive only non-EDI data */
char Nonedionly[EXP_MAXLEN_RV_NONEDIONLY];

/* The option to receive only EDI data */
char Edionly[EXP_MAXLEN_RV_EDIONLY];

/* The option to overwrite a file */
char Overwrite[EXP_MAXLEN_RV_OVERWRITE];

/* The time to wait for data */
char Waittime[EXP_MAXLEN_RV_WAITTIME];

/* The message key */
char Msgkey[EXP_MAXLEN_RV_MSGKEY];

/* The start date and time */
char StartDate[EXP_MAXLEN_RV_START_DATE];
/* The end date and time */
char EndDate[EXP_MAXLEN_RV_END_DATE];

/* The time zone */
char GlTimeZone[EXP_MAXLEN_RV_GL_TIME_ZONE];

/* The translation table */
char XlateTbl[EXP_MAXLEN_RV_XLATE_TBL];

/* The option to receive data as EDI */
char Receiveedi[EXP_MAXLEN_RV_RECEIVEEDI];

};
Input field definitions for ExpAddRecvOrdrReqStruct

For a list of the output fields, see “Output field definitions for ExpRecvOrdrAddedRspStruct” on page 293.

**ProjName**
The project name. Because you may have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**OrdrName**
The order name for a receive command.

- Max length: EXP_MAXLEN_RV_ORDR_NAME (20)
- Min length: EXP_MINLEN_RV_ORDR_NAME (1)
- Unique: Yes, because this is a key field.
- Required: Yes
- Case sensitive: No

**OrderDescr**
The description of the order.

- Max length: EXP_MAXLEN_RV_ORDER_DESCR (79)
- Min length: EXP_MINLEN_RV_ORDER_DESCR (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: Yes, for printable data.
**IeSysid**
The Information Exchange system where the account ID and user ID are registered. If you specify IeSysid, you must specify IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_DS_IE_SYSID (3)
- Min length: EXP_MINLEN_DS_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- Max length: EXP_MAXLEN_RV_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_RV_IE_MBOX_ID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**TableType**
The alias-table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_RV_TABLE_TYPE (1)
- Min length: EXP_MINLEN_RV_TABLE_TYPE (1)
- Default value: Blank
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - G  Alias table for global account.
  - O  Alias table for a specific account.
  - P  Alias table for private use.
**AliasTbl**

The alias-table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

- **Max length:** EXP_MAXLEN_RV_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_RV_ALIAS_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Aliasname**

This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_RV_ALIASNAME (16)
- **Min length:** EXP_MINLEN_RV_ALIASNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Listname**

The name of the list.

- **Max length:** EXP_MAXLEN_RV_LISTNAME (8)
- **Min length:** EXP_MINLEN_RV_LISTNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddRecvOrdrReqStruct

**Nickname**
This is a nickname from the address book. Use Nickname with AddrType to determine the network address to use for sending or receiving. If you specify Nickname, you must specify AddrType.

- **Max length:** EXP_MAXLEN_RV_NICKNAME (20)
- **Min length:** EXP_MINLEN_RV_NICKNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**AddrType**
The address type to use when sending or receiving data. Use AddrType with Nickname to determine the network address. If you specify AddrType, you must specify Nickname.

- **Max length:** EXP_MAXLEN_RV_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_RV_ADDR_TYPE (1)
- **Default value:** RM_DFLT_RV_ADDR_TYPE (E)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **A** Use the alias name and the alias table.
  - **D** Use a distribution list with the name specified in the Nickname field.
  - **E** Use the Information Exchange address.
  - **X** Use the Mail Exchange address.

**MsgClass**
The Information Exchange user message class specified by the user.

- **Max length:** EXP_MAXLEN_RV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_RV_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Fieldid**
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_RV_FILEID (65)
- **Min length:** EXP_MINLEN_RV_FILEID (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, as a system file path.

**Archiveid**
The archival reference identifier the user wants Information Exchange to assign to the received files.

- **Max length:** EXP_MAXLEN_RV_ARCHIVEID (8)
- **Min length:** EXP_MINLEN_RV_ARCHIVEID (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Allfiles**
The option to receive all files that match the RECEIVE specifications or just the first file in the mailbox that matches the RECEIVE specifications.

- **Max length:** EXP_MAXLEN_RV_ALLFILES (1)
- **Min length:** EXP_MINLEN_RV_ALLFILES (1)
- **Default value:** RM_DFLT_RV_ALLFILES (Y)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Receive first matching file.
  - **Y** Receive all matching files.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddRecvOrdrReqStruct

**Multfiles**
This determines whether the files received are placed into separate files or are concatenated into one file.

- **Max length:** EXP_MAXLEN_RV_MULTFILES_OPT (1)
- **Min length:** EXP_MINLEN_RV_MULTFILES_OPT (1)
- **Default value:** RM_DFLT_RV_MULTFILES_OPT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Store the data in a single file.
  - **Y** Store the received files into individual files. Use the file masks specified to generate new file names, if necessary.

**Origfile**
This indicates whether the user wants to receive data into a file, using the original name from the sending system.

- **Max length:** EXP_MAXLEN_RV_ORIGFILE (1)
- **Min length:** EXP_MINLEN_RV_ORIGFILE (1)
- **Default value:** RM_DFLT_RV_ORIGFILE (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not use the original file name.
  - **Y** Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.
Input field definitions for ExpAddRecvOrdrReqStruct

**Format**
This indicates whether or not to receive data as a file or as e-mail.

- **Max length:** EXP_MAXLEN_RV_FORMAT (1)
- **Min length:** EXP_MINLEN_RV_FORMAT (1)
- **Default value:** RM_DFLT_RV_FORMAT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not format as e-mail.
  - **Y** Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.

**Recordsize**
This indicates whether the EDI data is broken up into fixed-length records by placing CRLF characters in the data at specified intervals.

- **Max length:** EXP_MAXLEN_RV_RECORDSIZE (3)
- **Min length:** EXP_MINLEN_RV_RECORDSIZE (1)
- **Max value:** EXP_MAXVAL_RV_RECORDSIZE (999)
- **Min value:** EXP_MINVAL_RV_RECORDSIZE (0)
- **Default value:** RM_DFLT_RV_RECORDSIZE (0)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, as numeric data.

**Processlen**
This controls the processing of length delimiters at the beginning of each record.

- **Max length:** EXP_MAXLEN_RV_PROCESSLEN (1)
- **Min length:** EXP_MINLEN_RV_PROCESSLEN (1)
- **Default value:** RM_DFLT_RV_PROCESSLEN (C)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **C** Convert the line-length delimiters to CRLF delimiters.
  - **I** Ignore line-length delimiters.
  - **R** Remove line-length delimiters.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddRecvOrdrReqStruct

**Autoedi**
This indicates whether EDI file processing is automatic if the CDH indicates EDI format.

- **Max length:** EXP_MAXLEN_RV_AUTOEDI (1)
- **Min length:** EXP_MINLEN_RV_AUTOEDI (1)
- **Default value:** RM_DFLT_RV_AUTOEDI (Y)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - Y Automatically process the data as EDI.
  - N Do not process the data as EDI.

**Ediopt**
This indicates whether CRLF characters are added after each segment delimiter in the EDI file.

- **Max length:** EXP_MAXLEN_RV_EDIOPT (1)
- **Min length:** EXP_MINLEN_RV_EDIOPT (1)
- **Default value:** RM_DFLT_RV_EDIOPT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Do not add CRLF at the end of EDI segments.
  - Y Add CRLF characters at the end of EDI segments.
  - B Add CRLF characters after the number of bytes specified in Recordsize.

**Reserved1**
This is a reserved field; leave it blank.
Length = 1.
**Requeued**
This indicates whether to receive requeued files (files retrieved from archive into the mailbox).

- **Max length:** EXP_MAXLEN_RV_REQUEUED (1)
- **Min length:** EXP_MINLEN_RV_REQUEUED (1)
- **Default value:** RM_DFLT_RV_REQUEUED (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not receive requeued files.
  - **Y** Receive requeued files. Do not specify a destination on the receive command if you specify this option.

**Removeeof**
This indicates whether EOF characters are removed from the end of received files.

- **Max length:** EXP_MAXLEN_RV_REMOVEEOF (1)
- **Min length:** EXP_MINLEN_RV_REMOVEEOF (1)
- **Default value:** RM_DFLT_RV_REMOVEEOF (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not remove EOF characters.
  - **Y** Remove EOF characters.

**Nonedionly**
This specifies whether to receive only non-EDI data.

- **Max length:** EXP_MAXLEN_RV_NONEDIONLY (1)
- **Min length:** EXP_MINLEN_RV_NONEDIONLY (1)
- **Default value:** RM_DFLT_RV_NONEDIONLY (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Receive all data.
  - **Y** Check the CDH and receive only non-EDI data.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddRecvOrdReqStruct

**Edionly**
This specifies whether or not to receive only EDI data.

- **Max length:** EXP_MAXLEN_RV_EDIONLY (1)
- **Min length:** EXP_MINLEN_RV_EDIONLY (1)
- **Default value:** RM_DFLT_RV_EDIONLY (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Receive all data.
  - **Y** Check the CDH and receive only EDI data.

**Overwrite**
The action to take when receiving a file that already exists.

- **Max length:** EXP_MAXLEN_RV_OVERWRITE (1)
- **Min length:** EXP_MINLEN_RV_OVERWRITE (1)
- **Default value:** RM_DFLT_RV_OVERWRITE (A)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **A** Append the new data to the existing file.
  - **G** Generate a new file name using the file mask specified for the new data.
  - **O** Overwrite the existing file.

**Waittime**
The number of minutes Expedite for Windows should wait to receive data.

- **Max length:** EXP_MAXLEN_RV_WAITTIME (2)
- **Min length:** EXP_MINLEN_RV_WAITTIME (1)
- **Min value:** EXP_MAXVAL_RV_WAITTIME (0)
- **Max value:** EXP_MINVAL_RV_WAITTIME (9)
- **Default value:** RM_DFLT_RV_WAITTIME (0)
- **Required:** No
- **Validation:** Yes, as numeric data between 0 - 9.
**Msgkey**
The unique message key you use to receive a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_RV_MSGKEY (20)
- **Min length:** EXP_MINLEN_RV_MSGKEY (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**StartDate**
The starting date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_START_DATE (14)
- **Min length:** EXP_MINLEN_RV_START_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for date and time in YYYYMMDDHHMMSS format.

**EndDate**
The ending date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_END_DATE (14)
- **Min length:** EXP_MINLEN_RV_END_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for date and time in YYYYMMDDHHMMSS format.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddRecvOrdrReqStruct

TimeZone
The time zone specified in the TimeZone field of the dropoff box or ExpStartSess command.

Max length: EXP_MAXLEN_RV_GL_TIME_ZONE (5)
Min length: EXP_MINLEN_RV_GL_TIME_ZONE (1)
Default value: L
Required: No
Case sensitive: No
Validation: The valid values are:
  G  Greenwich mean time (GMT)
  L  Local time.

XlateTbl
The translation table name.

Max length: EXP_MAXLEN_RV_XLATE_TBL (8)
Min length: EXP_MINLEN_RV_XLATE_TBL (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: Yes, it must contain the file name for the translation table without the XLT extension.

Receiveedi
The option to receive data as EDI or unformatted.

Max length: EXP_MAXLEN_RV_RECEIVEEDI (1)
Min length: EXP_MINLEN_RV_RECEIVEEDI (1)
Default value: No
Required: No
Case sensitive: Yes
Validation: The valid values are:
  N  Receive as unformatted data.
  Y  Receive as EDI data.
Output arguments for ExpAddRecvOrdr

**struct ExpRecvOrdrAddedRspStruct**

{
 /* The order name for a receive */
 char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];
 /* The order description */
 char OrderDescr[EXP_MAXLEN_RV_ORDER_DESCR];
 /* The Information Exchange system ID */
 char IeSysid[EXP_MAXLEN_RV_IE_SYSID];
 /* The Information Exchange account ID and user ID */
 char IeMboxID[EXP_MAXLEN_RV_IE_MBOX_ID];
 /* The table type for the alias table */
 char TableType[EXP_MAXLEN_RV_TABLE_TYPE];
 /* The Information Exchange alias table */
 char AliasTbl[EXP_MAXLEN_RV_ALIAS_TBL];
 /* The Information Exchange alias name */
 char Aliasname[EXP_MAXLEN_RV_ALIASNAME];
 /* The list name */
 char Listname[EXP_MAXLEN_RV_LISTNAME];
 /* The nickname */
 char Nickname[EXP_MAXLEN_RV_NICKNAME];
 /* The address type */
 char AddrType[EXP_MAXLEN_RV_ADDR_TYPE];
 /* The Information Exchange message class */
 char MsgClass[EXP_MAXLEN_RV_MSG_CLASS];
 /* The file name for the received data */
 char Fileid[EXP_MAXLEN_RV_FILEID];
 /* The archive ID of the received data */
 char Archiveid[EXP_MAXLEN_RV_ARCHIVEID];
 /* The option to receive one file or all files that match */
 char Allfiles[EXP_MAXLEN_RV_ALLFILES];
 /* The multiple files parameter for the receive */
 char Multfiles[EXP_MAXLEN_RV_MULTFILES];
/* The receipt of files with the original name */
char Origfile[EXP_MAXLEN_RV_ORIGFILE];

/* The format for the received data */
char Format[EXP_MAXLEN_RV_FORMAT];

/* The EDI record size */
char Recordsize[EXP_MAXLEN_RV_RECORDSIZE];

/* The processing for line length delimiters */
char Processlen[EXP_MAXLEN_RV_PROCESSLEN];

/* The automatic processing for EDI data */
char Autoedi[EXP_MAXLEN_RV_AUTOEDI];

/* The options for EDI data processing */
char Ediopt[EXP_MAXLEN_RV_EDIOPT];

/* Reserved field */
char Reserved1[20];

/* The requeued parameter for a receive */
char Requeued[EXP_MAXLEN_RV_REQUEUED];

/* The controls for the removal of EOF characters */
char Removeeof[EXP_MAXLEN_RV_REMOVEEOF];

/* The option to receive only non-EDI data */
char Nonedionly[EXP_MAXLEN_RV_NONEDIONLY];

/* The option to receive only EDI data */
char Edionly[EXP_MAXLEN_RV_EDIONLY];

/* The option to overwrite a file */
char Overwrite[EXP_MAXLEN_RV_OVERWRITE];

/* The time to wait for data */
char Waittime[EXP_MAXLEN_RV_WAITTIME];

/* The message key */
char Msgkey[EXP_MAXLEN_RV_MSGKEY];

/* The start date and time */
char StartDate[EXP_MAXLEN_RV_START_DATE];

/* The end date and time */
char EndDate[EXP_MAXLEN_RV_END_DATE];
Output arguments for \textit{ExpAddRecvOrdr}

/* The time zone */
char GlTimeZone[\texttt{EXP\_MAXLEN\_RV\_GL\_TIME\_ZONE}];

/* The translation table */
char XlateTbl[\texttt{EXP\_MAXLEN\_RV\_XLATE\_TBL}];

/* The option to receive data as EDI */
char Receiveedi[\texttt{EXP\_MAXLEN\_RV\_RECEIVEEDI}];
};
Output field definitions for ExpRecvOrdrAddedRspStruct

The output can include any or all of these fields.

**OrdrName**
The order name for a receive command.

- **Max length:** EXP_MAXLEN_RV_ORDR_NAME (20)
- **Min length:** EXP_MINLEN_RV_ORDR_NAME (1)
- **Contents:** Printable data.

**OrderDescr**
The description of the order.

- **Max length:** EXP_MAXLEN_RV_ORDER_DESCR (79)
- **Min length:** EXP_MINLEN_RV_ORDER_DESCR (1)
- **Contents:** Printable data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_RV_IE_SYSID (3)
- **Min length:** EXP_MINLEN_RV_IE_SYSID (1)
- **Contents:** Printable data.

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- **Max length:** EXP_MAXLEN_RV_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_RV_IE_MBOX_ID (1)
- **Contents:** Printable data.
### TableType
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

<table>
<thead>
<tr>
<th>Max length</th>
<th>EXP_MAXLEN_RV_TABLE_TYPE (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_TABLE_TYPE (1)</td>
</tr>
<tr>
<td>Contents</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td></td>
<td>G    Alias table for global account.</td>
</tr>
<tr>
<td></td>
<td>O    Alias table for a specific account.</td>
</tr>
<tr>
<td></td>
<td>P    Alias table for private use.</td>
</tr>
</tbody>
</table>

### AliasTbl
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

<table>
<thead>
<tr>
<th>Max length</th>
<th>EXP_MAXLEN_RV_ALIAS_TBL (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_ALIAS_TBL (1)</td>
</tr>
<tr>
<td>Contents</td>
<td>Printable data.</td>
</tr>
</tbody>
</table>

### Aliasname
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

<table>
<thead>
<tr>
<th>Max length</th>
<th>EXP_MAXLEN_RV_ALIASNAME (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_ALIASNAME (1)</td>
</tr>
<tr>
<td>Contents</td>
<td>Printable data.</td>
</tr>
</tbody>
</table>

### Listname
The name of the list.

<table>
<thead>
<tr>
<th>Max length</th>
<th>EXP_MAXLEN_RV_LISTNAME (8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_LISTNAME (1)</td>
</tr>
<tr>
<td>Contents</td>
<td>Printable data.</td>
</tr>
</tbody>
</table>

### Nickname
The nickname from the address book. The Nickname is specified with AddrType to determine the network address for sending and receiving.

<table>
<thead>
<tr>
<th>Max length</th>
<th>EXP_MAXLEN_RV_NICKNAME (20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_NICKNAME (1)</td>
</tr>
<tr>
<td>Contents</td>
<td>Printable data.</td>
</tr>
</tbody>
</table>
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvOrdrAddedRspStruct

### AddrType
The address type to use when sending or receiving data. The AddrType is specified with Nickname to determine the network address.

- **Max length:** EXP_MAXLEN_RV_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_RV_ADDR_TYPE (1)
- **Contents:** One of these valid values:
  - A Use the alias name and the alias table.
  - D Use a distribution list with the name specified in the Nickname field.
  - E Use the Information Exchange address.
  - X Use the Mail Exchange address.

### MsgClass
The Information Exchange user message class specified by the user.

- **Max length:** EXP_MAXLEN_RV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_RV_MSG_CLASS (1)
- **Contents:** Printable data.

### Fileid
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_RV_FILEID (65)
- **Min length:** EXP_MINLEN_RV_FILEID (1)
- **Required:** Yes
- **Case sensitive:** No
- **Contents:** Printable data.

### Archiveid
The archival reference identifier the user wants Information Exchange to assign to the received files.

- **Max length:** EXP_MAXLEN_RV_ARCHIVEID (8)
- **Min length:** EXP_MINLEN_RV_ARCHIVEID (1)
- **Contents:** Printable data.
**Allfiles**

The option to receive all files that match the RECEIVE specifications or just the first file in the mailbox that matches the RECEIVE specifications.

- **Max length:** EXP_MAXLEN_RV_ALLFILES (1)
- **Min length:** EXP_MINLEN_RV_ALLFILES (1)
- **Contents:** The valid values are:
  - N Receive first matching file.
  - Y Receive all matching files.

**Multfiles**

This determines whether the files the user receives are placed into separate files or are concatenated into one file.

- **Max length:** EXP_MAXLEN_RV_MULTFILES_OPT (1)
- **Min length:** EXP_MINLEN_RV_MULTFILES_OPT (1)
- **Contents:** One of these valid values:
  - N Store the data in a single file.
  - Y Store the received files into individual files. Use the file masks specified to generate new file names, if necessary.

**Origfile**

This indicates whether the user wants to receive data into a file, using the original name from the sending system.

- **Max length:** EXP_MAXLEN_RV_ORIGFILE (1)
- **Min length:** EXP_MINLEN_RV_ORIGFILE (1)
- **Contents:** One of these valid values:
  - N Do not use the original file name.
  - Y Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.
Format
This indicates whether or not to receive data as a file or as email.

Max length: \text{EXP\_MAXLEN\_RV\_FORMAT} (1)
Min length: \text{EXP\_MINLEN\_RV\_FORMAT} (1)
Contents: One of these valid values:
\begin{itemize}
  \item \text{N} Do not format as e-mail.
  \item \text{Y} Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.
\end{itemize}

Recordsize
This indicates whether the EDI data is broken into fixed-length records by placing CRLF characters in the data at specified intervals.

Max length: \text{EXP\_MAXLEN\_RV\_RECORDSIZE} (3)
Min length: \text{EXP\_MINLEN\_RV\_RECORDSIZE} (1)
Max value: \text{EXP\_MAXVAL\_RV\_RECORDSIZE} (999)
Min value: \text{EXP\_MINVAL\_RV\_RECORDSIZE} (0)
Contents: Numeric data.

Processlen
This controls the processing of length delimiters at the beginning of each record.

Max length: \text{EXP\_MAXLEN\_RV\_PROCESSLEN} (1)
Min length: \text{EXP\_MINLEN\_RV\_PROCESSLEN} (1)
Contents: One of these valid values:
\begin{itemize}
  \item \text{C} Convert the line-length delimiters to CRLF delimiters.
  \item \text{I} Ignore line-length delimiters.
  \item \text{R} Remove line-length delimiters.
\end{itemize}

Autoedi
This indicates whether EDI file processing is automatic if the CDH indicates EDI format.

Max length: \text{EXP\_MAXLEN\_RV\_AUTOEDI} (1)
Min length: \text{EXP\_MINLEN\_RV\_AUTOEDI} (1)
Contents: One of these valid values:
\begin{itemize}
  \item \text{Y} Automatically process the data as EDI.
  \item \text{N} Do not process the data as EDI.
Output field definitions for ExpRecvOrdrAddedRspStruct

**Ediopt**
This indicates whether CRLF characters are added after each segment delimiter in the EDI file.

- **Max length:** EXP_MAXLEN_RV_EDIOPT (1)
- **Min length:** EXP_MINLEN_RV_EDIOPT (1)
- **Contents:** One of these valid values:
  - **N** Do not add CRLF at the end of EDI segments.
  - **Y** Add CRLF characters at the end of EDI segments.
  - **B** Add CRLF characters after the number of bytes specified in Recordsize.

**Reserved1**
This is a reserved field; leave it blank.
Length = 1.

**Requeued**
This indicates whether to receive requeued files (files retrieved from archive into the mailbox).

- **Max length:** EXP_MAXLEN_RV_REQUEUED (1)
- **Min length:** EXP_MINLEN_RV_REQUEUED (1)
- **Contents:** One of these valid values:
  - **N** Do not receive requeued files.
  - **Y** Receive requeued files. Do not specify a destination on the receive command if you specify this option.

**Removeeof**
This indicates whether EOF characters are removed from the end of received files.

- **Max length:** EXP_MAXLEN_RV_REMOVEEOF (1)
- **Min length:** EXP_MINLEN_RV_REMOVEEOF (1)
- **Contents:** One of these valid values:
  - **N** Do not remove EOF characters.
  - **Y** Remove EOF characters.
Nonedionly
This specifies whether to receive only non-EDI data.

Max length: EXP_MAXLEN_RV_NONEDIONLY (1)
Min length: EXP_MINLEN_RV_NONEDIONLY (1)
Contents: One of these valid values:
N Receive all data.
Y Check the CDH and receive only non-EDI data.

Edionly
This specifies whether or not to receive only EDI data.

Max length: EXP_MAXLEN_RV_EDIONLY (1)
Min length: EXP_MINLEN_RV_EDIONLY (1)
Contents: One of these valid values:
N Receive all data.
Y Check the CDH and receive only EDI data.

Overwrite
The action to take when receiving a file that already exists.

Max length: EXP_MAXLEN_RV_OVERWRITE (1)
Min length: EXP_MINLEN_RV_OVERWRITE (1)
Contents: One of these valid values:
A Append the new data to the existing file.
G Generate a new file name using the file mask specified for the new data.
O Overwrite the existing file.

Waittime
The number of minutes Expedite for Windows should wait to receive data.

Max length: EXP_MAXLEN_RV_WAITTIME (1)
Min length: EXP_MINLEN_RV_WAITTIME (1)
Min value: EXP_MAXVAL_RV_WAITTIME (0)
Max value: EXP_MINVAL_RV_WAITTIME (9)
Contents: Numeric data between 0 - 9.
**Msgkey**
The unique message key you use to receive a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_RV_MSGKEY (20)
- **Min length:** EXP_MINLEN_RV_MSGKEY (1)
- **Contents:** Printable data.

**StartDate**
The starting date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_START_DATE (14)
- **Min length:** EXP_MINLEN_RV_START_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**EndDate**
The ending date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_END_DATE (14)
- **Min length:** EXP_MINLEN_RV_END_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**GlTimeZone**
Indicates whether the date and time field specified on the receive order is local time (as specified in the TimeZone field on the dropoff box or `ExpStartSess` function) or Greenwich mean time.

- **Max length:** EXP_MAXLEN_RV_GL_TIME_ZONE (1)
- **Min length:** EXP_MINLEN_RV_GL_TIME_ZONE (1)
- **Contents:** One of these valid values:
  - G Greenwich mean time (GMT).
  - L Local time.

**XlateTbl**
The translation table name.

- **Max length:** EXP_MAXLEN_RV_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_RV_XLATE_TBL (1)
- **Contents:** The file name of the translation table without the XLT extension.
Receiveedi
The option to receive data as EDI or unformatted.

Max length: EXP_MAXLEN_RV_RECEIVEEDI (1)
Min length: EXP_MINLEN_RV_RECEIVEEDI (1)
Contents: One of these valid values:
  N  Receive as unformatted data.
  Y  Receive as EDI data.
**ExpAsgRecvOrdrToDropBox**

This function assigns the receive order to the dropoff box. If the receive order name or the dropoff box specified does not exist, the return code is set to 8. If the receive order is already assigned to the dropoff box, the return code is set to 4.

**Function prototype ExpAsgRecvOrdrToDropBox**

```c
int ExpAsgRecvOrdrToDropBox(
    struct ExpAsgRecvOrdrToDropBoxReqStruct * Input
);
```

**Input arguments for ExpAsgRecvOrdrToDropBox**

```c
struct ExpAsgRecvOrdrToDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The receive order name */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];
};
```
There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

| Max length: | EXP_MAXLEN_PJ_PROJ_NAME (20) |
| Min length: | EXP_MINLEN_PJ_PROJ_NAME (1) |
| Required:   | Yes                          |
| Case sensitive: | No            |
| Validation: | Yes, the project must be open. |

**OrdrName**
The receive order name.

| Max length: | EXP_MAXLEN_RV_ORDR_NAME (20) |
| Min length: | EXP_MINLEN_RV_ORDR_NAME (1) |
| Required:   | Yes                          |
| Case sensitive: | No            |
| Validation: | Yes, the receive order name must exist in the database. |

**DropBoxName**
The dropoff box name.

| Max length: | EXP_MAXLEN_DR_DROP_BOX_NAME (20) |
| Min length: | EXP_MINLEN_DR_DROP_BOX_NAME (1) |
| Required:   | Yes                          |
| Case sensitive: | No            |
| Validation: | Yes, the dropoff box name must exist in the database. |
ExpDeleteRecvOrdr

This function deletes the receive order from the MailBox database. If the receive order name specified does not exist, the return code is set to 4.

If the receive order is assigned to a dropoff box, it is unassigned before it is deleted. If this receive order was the last one assigned to the dropoff box, the dropoff box remains in the MailBox database.

Function prototype for ExpDeleteRecvOrdr

```c
int ExpDeleteRecvOrdr(
    struct ExpDeleteRecvOrdrReqStruct * Input
);
```

Input arguments for ExpDeleteRecvOrdr

```c
struct ExpDeleteRecvOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The receive order name */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];
};
```
Input field definitions for ExpDeleteRecvOrdrReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**OrdrName**
The receive order name.

- Max length: EXP_MAXLEN_RV_ORDR_NAME (20)
- Min length: EXP_MINLEN_RV_ORDR_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the receive order must exist in the database.
ExpListRecvOrdr

This function lists the receive orders on the order shelf in the Mailbox database that correspond to the query that is formed in the request structure. If the query results in no receive orders found, the return code is set to 4.

Function prototype for ExpListRecvOrdr

```c
int ExpListRecvOrdr(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpRecvOrdrListedRspStruct Output[]
);
```

Input arguments for ExpListRecvOrdr

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char startIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListRecvOrdrReqStruct

For a list of the output fields, see “Output field definitions for ExpRecvOrdrListedRspStruct” on page 313.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**SearchParm**
The field on which to search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (15)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the search fields are:
  - Name
  - Description
  - MailboxId
  - FileId
  - Class

**SearchOp**
The operator to use for the search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_OPER (2)
- **Min length:** EXP_MINLEN_SEARCH_OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** Yes, the valid characters are: =, <, >, <=, or >=.
**SearchValue**
The search value to use.

Max length: EXP_MAXLEN_CF_SEARCH_VALUE (25)
Min length: EXP_MINLEN_SEARCH_VALUE (1)
Required: Yes, if a SearchParm is specified.
Case sensitive: Yes
Validation: Yes, for printable data.

**SortParm**
The field on which to sort.

Max length: EXP_MAXLEN_CF_SORT_PARM (15)
Min length: EXP_MINLEN_SORT_PARM (1)
Required: Yes, if a SortOrder is specified.
Case sensitive: No
Validation: Yes, the sort fields are:
- Name
- Description
- MailboxId
- Fileid
- Class

**SortOrder**
The order in which to sort the list.

Max length: EXP_MAXLEN_CF_SORT_ORDER (1)
Min length: EXP_MINLEN_SORT_ORDER (1)
Required: No
Case sensitive: Yes
Validation: Yes, the valid values are:
- **RM_SORT_ASCENDING (A)**
  Sort the list in ascending order.
- **RM_SORT_DESCENDING (D)**
  Sort the list in descending order.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListRecvOrdReqStruct

NumEntries
The number of entries to return in the response structure.

- **Max length**: EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length**: EXP_MINLEN_NUM_ENTRIES (1)
- **Max value**: EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value**: EXP_MINVAL_NUM_ENTRIES (1)
- **Required**: Yes
- **Case sensitive**: No
- **Validation**: Yes, for numeric data.

StartIndex
The entry number at which to start the search.

- **Max length**: EXP_MAXLEN_CF_START_INDEX (5)
- **Min length**: EXP_MINLEN_START_INDEX (1)
- **Max value**: EXP_MAXVAL_START_INDEX (65535)
- **Min value**: EXP_MINVAL_START_INDEX (1)
- **Required**: No
- **Case sensitive**: No
- **Validation**: Yes, for numeric data.
Output arguments for ExpListRecvOrdr

```c
struct ExpRecvOrdrListedRspStruct
{
    /* The order type for sending */
    char OrdrType[EXP_MAXLEN_DR_ORDR_TYPE];

    /* The order name for a receive */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];

    /* The order description */
    char OrderDescr[EXP_MAXLEN_RV_ORDER_DESCR];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_RV_IE_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxID[EXP_MAXLEN_RV_IE_MBOX_ID];

    /* The table type for the alias table */
    char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_RV_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_RV_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_RV_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_RV_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_RV_ADDR_TYPE];

    /* The Information Exchange message class */
    char MsgClass[EXP_MAXLEN_RV_MSG_CLASS];

    /* The file name for the received data */
    char Fileid[EXP_MAXLEN_RV_FILEID];

    /* The archive ID of the received data */
    char Archiveid[EXP_MAXLEN_RV_ARCHIVEID];
};
```
/* The option to receive one file or all files that match */
char Allfiles[EXP_MAXLEN_RV_ALLFILES];

/* The multiple files parameter for the receive */
char Multfiles[EXP_MAXLEN_RV_MULTFILES];

/* The receipt of files with the original name */
char Origfile[EXP_MAXLEN_RV_ORIGFILE];

/* The format for the received data */
char Format[EXP_MAXLEN_RV_FORMAT];

/* The EDI record size */
char Recordsize[EXP_MAXLEN_RV_RECORDSIZE];

/* The processing for line length delimiters */
char Processlen[EXP_MAXLEN_RV_PROCESSLEN];

/* The automatic processing for EDI data */
char Autoedi[EXP_MAXLEN_RV_AUTOEDI];

/* The option for EDI data processing */
char Ediopt[EXP_MAXLEN_RV_EDIOPT];

/* Reserved field */
char Reserved1[20];

/* The requeued parameter for a receive */
char Requeued[EXP_MAXLEN_RV_REQUEUED];

/* The controls for the removal of EOF characters */
char Removeeof[EXP_MAXLEN_RV_REMOVEEOF];

/* The option to receive only non-EDI data */
char Nonedionly[EXP_MAXLEN_RV_NONEDIONLY];

/* The option to receive only EDI data */
char Edionly[EXP_MAXLEN_RV_EDIONLY];

/* The option to overwrite files */
char Overwrite[EXP_MAXLEN_RV_OVERWRITE];

/* The time to wait for data */
char Waittime[EXP_MAXLEN_RV_WAITTIME];

/* The message key */
char Msgkey[EXP_MAXLEN_RV_MSGKEY];
Output arguments for ExpListRecvOrdr

/* The start date and time */
char StartDate[EXP_MAXLEN_RV_START_DATE];

/* The end date and time */
char EndDate[EXP_MAXLEN_RV_END_DATE];

/* The time zone */
char GlTimeZone[EXP_MAXLEN_RV_GL_TIME_ZONE];

/* The translation table */
char XlateTbl[EXP_MAXLEN_RV_XLATE_TBL];

/* The option to receive data as EDI or unformatted */
char Receiveedi[EXP_MAXLEN_RV_RECEIVEEDI];

};
Output field definitions for ExpRecvOrdrListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of receive orders copied into the output structure.

**OrdrType**
The order type identifier for a receive command.

  - Max length: EXP_MAXLEN_DR_ORDR_TYPE (1)
  - Min length: EXP_MINLEN_DR_ORDR_TYPE (1)
  - Contents: The character R.

**OrdrName**
The order name for a receive command.

  - Max length: EXP_MAXLEN_RV_ORDR_NAME (20)
  - Min length: EXP_MINLEN_RV_ORDR_NAME (1)
  - Contents: Printable data.

**OrderDescr**
The description of the order.

  - Max length: EXP_MAXLEN_RV_ORDER_DESCR (79)
  - Min length: EXP_MINLEN_RV_ORDER_DESCR (1)
  - Contents: Printable data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

  - Max length: EXP_MAXLEN_RV_IE_SYSID (3)
  - Min length: EXP_MINLEN_RV_IE_SYSID (1)
  - Contents: Printable data.
**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

Max length: \( \text{EXP\_MAXLEN\_RV\_IE\_MBOX\_ID} (16) \)
Min length: \( \text{EXP\_MINLEN\_RV\_IE\_MBOX\_ID} (1) \)
Contents: Printable data.

**TableType**
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

Max length: \( \text{EXP\_MAXLEN\_RV\_TABLE\_TYPE} (1) \)
Min length: \( \text{EXP\_MINLEN\_RV\_TABLE\_TYPE} (1) \)
Contents: One of these valid values:

- **G** Alias table for global account.
- **O** Alias table for a specific account.
- **P** Alias table for private use.

**AliasTbl**
The alias-table name of an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

Max length: \( \text{EXP\_MAXLEN\_RV\_ALIAS\_TBL} (3) \)
Min length: \( \text{EXP\_MINLEN\_RV\_ALIAS\_TBL} (1) \)
Contents: Printable data.

**Aliasname**
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

Max length: \( \text{EXP\_MAXLEN\_RV\_ALIASNAME} (16) \)
Min length: \( \text{EXP\_MINLEN\_RV\_ALIASNAME} (1) \)
Contents: Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvOrdrListedRspStruct

**Listname**
The name of the list.

- **Max length:** EXP_MAXLEN_RV_LISTNAME (8)
- **Min length:** EXP_MINLEN_RV_LISTNAME (1)
- **Contents:** Printable data.

**Nickname**
The nickname from the address book. The Nickname is specified along with AddrType to determine the network address for sending and receiving.

- **Max length:** EXP_MAXLEN_RV_NICKNAME (20)
- **Min length:** EXP_MINLEN_RV_NICKNAME (1)
- **Contents:** Printable data.

**AddrType**
The address type to use for sending or receiving data. The AddrType is specified along with Nickname to determine the network address.

- **Max length:** EXP_MAXLEN_RV_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_RV_ADDR_TYPE (1)
- **Contents:** One of these valid values:
  - **A** Use the alias name and the alias table.
  - **D** Use a distribution list with the name specified in the Nickname field.
  - **E** Use the Information Exchange address.
  - **X** Use the Mail Exchange address.

**MsgClass**
The Information Exchange user message class specified by the user.

- **Max length:** EXP_MAXLEN_RV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_RV_MSG_CLASS (1)
- **Contents:** Printable data.

**Fileid**
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_RV_FILEID (65)
- **Min length:** EXP_MINLEN_RV_FILEID (1)
- **Contents:** A system file path.
Archiveid
The archival reference identifier the user wants Information Exchange to assign to the received files.

Max length: EXP_MAXLEN_RV_ARCHIVEID (8)
Min length: EXP_MINLEN_RV_ARCHIVEID (1)
Contents: Printable data.

Allfiles
This indicates whether all the files that match the RECEIVE specifications are received or just the first file in the mailbox that matches the RECEIVE specifications.

Max length: EXP_MAXLEN_RV_ALLFILES (1)
Min length: EXP_MINLEN_RV_ALLFILES (1)
Contents: The valid values are:
N Receive first matching file.
Y Receive all matching files.

Multfiles
This determines whether the files the user receives are placed into separate files or are concatenated into one file.

Max length: EXP_MAXLEN_RV_MULTFILES_OPT (1)
Min length: EXP_MINLEN_RV_MULTFILES_OPT (1)
Contents: One of these valid values:
N Store the data in a single file.
Y Store the received files into individual files. Use the file masks specified to generate new file names, if necessary.

Origfile
This indicates whether the user wants to receive data into a file, using the original name from the sending system.

Max length: EXP_MAXLEN_RV_ORIGFILE (1)
Min length: EXP_MINLEN_RV_ORIGFILE (1)
Contents: One of these valid values:
N Do not use the original file name.
Y Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvOrdrListedRspStruct

**Format**
This indicates whether or not to receive data as a file or as e-mail.

Max length: EXP_MAXLEN_RV_FORMAT (1)  
Min length: EXP_MINLEN_RV_FORMAT (1)  
Contents: One of these valid values:
  - N  Do not format as e-mail.  
  - Y  Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.

**Recordsize**
This indicates whether the EDI data is broken up into fixed-length records by placing CRLF characters in the data at specified intervals.

Max length: EXP_MAXLEN_RV_RECORDSIZE (3)  
Min length: EXP_MINLEN_RV_RECORDSIZE (1)  
Max value: EXP_MAXVAL_RV_RECORDSIZE (999)  
Min value: EXP_MINVAL_RV_RECORDSIZE (0)  
Contents: Numeric data.

**Processlen**
This controls the processing of length delimiters at the beginning of each record.

Max length: EXP_MAXLEN_RV_PROCESSLEN (1)  
Min length: EXP_MINLEN_RV_PROCESSLEN (1)  
Contents: One of these valid values:
  - C  Convert the line-length delimiters to CRLF delimiters.  
  - I  Ignore line-length delimiters.  
  - R  Remove line-length delimiters.

**Autoedi**
This indicates whether EDI file processing is automatic if the CDH indicates EDI format.

Max length: EXP_MAXLEN_RV_AUTOEDI (1)  
Min length: EXP_MINLEN_RV_AUTOEDI (1)  
Contents: One of these valid values:
  - Y  Process the data as EDI automatically.  
  - N  Do not process the data as EDI.
Output field definitions for ExpRecvOrdrListedRspStruct

**Ediopt**
This indicates whether CRLF characters are added after each segment delimiter in the EDI file.

Max length: \( \text{EXP\_MAXLEN\_RV\_EDIOPT} \) (1)
Min length: \( \text{EXP\_MINLEN\_RV\_EDIOPT} \) (1)
Contents: One of these valid values:
- N: Do not add CRLF at the end of EDI segments.
- Y: Add CRLF characters at the end of EDI segments.
- B: Add CRLF characters after number of bytes specified in Recordsize.

**Reserved1**
This is a reserved field; leave it blank.
Length = 1.

**Requeued**
This indicates whether to receive requeued files (files retrieved from archive into the mailbox).

Max length: \( \text{EXP\_MAXLEN\_RV\_REQUEUED} \) (1)
Min length: \( \text{EXP\_MINLEN\_RV\_REQUEUED} \) (1)
Contents: One of these valid values:
- N: Do not receive requeued files.
- Y: Receive requeued files. Do not specify a destination on the receive command if you specify this option.

**Removeeof**
This indicates whether EOF characters are removed from the end of received files.

Max length: \( \text{EXP\_MAXLEN\_RV\_REMOVEEOF} \) (1)
Min length: \( \text{EXP\_MINLEN\_RV\_REMOVEEOF} \) (1)
Contents: One of these valid values:
- N: Do not remove EOF characters.
- Y: Remove EOF characters.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvOrdrListedRspStruct

**Nonedionly**
This specifies whether to receive only non-EDI data.

- **Max length:** EXP_MAXLEN_RV_NONEDIONLY (1)
- **Min length:** EXP_MINLEN_RV_NONEDIONLY (1)
- **Contents:** One of these valid values:
  - N  Receive all data.
  - Y  Check the CDH and receive only non-EDI data.

**Edionly**
This specifies whether or not to receive only EDI data.

- **Max length:** EXP_MAXLEN_RV_EDIONLY (1)
- **Min length:** EXP_MINLEN_RV_EDIONLY (1)
- **Contents:** One of these valid values:
  - N  Receive all data.
  - Y  Check the CDH and receive only EDI data.

**Overwrite**
The action to take when receiving a file that already exists.

- **Max length:** EXP_MAXLEN_RV_OVERWRITE (1)
- **Min length:** EXP_MINLEN_RV_OVERWRITE (1)
- **Contents:** One of these valid values:
  - A  Append the new data to the existing file.
  - G  Generate a new file name using the file mask specified for the new data.
  - O  Overwrite the existing file.

**Waittime**
The number of minutes Expedite for Windows should wait to receive data.

- **Max length:** EXP_MAXLEN_RV_WAITTIME (1)
- **Min length:** EXP_MINLEN_RV_WAITTIME (1)
- **Min value:** EXP_MAXVAL_RV_WAITTIME (0)
- **Max value:** EXP_MINVAL_RV_WAITTIME (9)
- **Contents:** Numeric data between 0 - 9.
**Msgkey**
The unique message key you use to receive a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_RV_MSGKEY (20)
- **Min length:** EXP_MINLEN_RV_MSGKEY (1)
- **Contents:** Printable data.

**StartDate**
The starting date and time you want to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_START_DATE (14)
- **Min length:** EXP_MINLEN_RV_START_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**EndDate**
The ending date and time you want to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_END_DATE (14)
- **Min length:** EXP_MINLEN_RV_END_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**GlTimeZone**
Indicates whether the date and time field specified on the receive order is local time (as specified in the TimeZone field on the dropoff box or **ExpStartSess** function) or Greenwich mean time.

- **Max length:** EXP_MAXLEN_RV_GL_TIME_ZONE (1)
- **Min length:** EXP_MINLEN_RV_GL_TIME_ZONE (1)
- **Contents:** One of these valid values:
  - G  Greenwich mean time (GMT).
  - L  Local time.

**XlateTbl**
The translation table name.

- **Max length:** EXP_MAXLEN_RV_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_RV_XLATE_TBL (1)
- **Contents:** Printable data.
**Receiveedi**
The option to receive data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_RV_RECEIVEEDI (1)
- **Min length:** EXP_MINLEN_RV_RECEIVEEDI (1)
- **Contents:** One of these valid values:
  - **N** Receive as unformatted data.
  - **Y** Receive as EDI data.
ExpReplaceRecvOrdr

This function replaces information about a receive order. If the receive order name specified does not exist, the return code is set to 8.

Function prototype ExpReplaceRecvOrdr

```c
int ExpReplaceRecvOrdr(
    struct ExpReplaceRecvOrdrReqStruct * Input,
    struct ExpRecvOrdrReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceRecvOrdr

```c
struct ExpReplaceRecvOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The order name for a receive */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];

    /* The file name for the received data */
    char Fileid[EXP_MAXLEN_RV_FILEID];

    /* The archive ID of received data */
    char Archiveid[EXP_MAXLEN_RV_ARCHIVEID];

    /* The option to receive one file or all files that match */
    char Allfiles[EXP_MAXLEN_RV_ALLFILES];

    /* The multiple files parameter for receive */
    char Multfiles[EXP_MAXLEN_RV_MULTFILES];

    /* The receipt of files with original names */
    char Origfile[EXP_MAXLEN_RV_ORIGFILE];

    /* The format for the data received */
    char Format[EXP_MAXLEN_RV_FORMAT];

    /* The EDI record size */
    char Recordsize[EXP_MAXLEN_RV_RECORDSIZE];

    /* The processing for LL delimiters */
    char Processlen[EXP_MAXLEN_RV_PROCESSLEN];

    /* The automatic processing for EDI data */
    char Autoedi[EXP_MAXLEN_RV_AUTOEDI];
}```
Chapter 4. Writing the interface for the MailBox database

Input arguments for ExpReplaceRecvOrdr

/* The options for EDI data processing */
char Ediopt[EXP_MAXLEN_RV_EDIOPT];

/* Reserved field */
char Reserved1[20];

/* The requeued parameter for receive */
char Requeued[EXP_MAXLEN_RV_REQUEUED];

/* This controls the removal of EOF characters */
char Removeeof[EXP_MAXLEN_RV_REMOVEEOF];

/* The option to receive only non-EDI data */
char Nonedionly[EXP_MAXLEN_RV_NONEDIONLY];

/* Specifies to receive only EDI data */
char Edionly[EXP_MAXLEN_RV_EDIONLY];

/* The option to overwrite, append, or generate a new filename */
char Overwrite[EXP_MAXLEN_RV_OVERWRITE];

/* The time to wait for data */
char Waittime[EXP_MAXLEN_RV_WAITTIME];

/* The order description */
char OrdrDescr[EXP_MAXLEN_RV_ORDR_DESCR];

/* The Information Exchange system ID */
char IeSysid[EXP_MAXLEN_RV_IE_SYSID];

/* The Information Exchange account ID */
char IeMboxId[EXP_MAXLEN_RV_IE_MBOX_ID];

/* The table type for the alias table */
char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

/* The Information Exchange alias table */
char AliasTbl[EXP_MAXLEN_RV_ALIAS_TBL];

/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_RV_ALIASNAME];

/* The list name */
char Listname[EXP_MAXLEN_RV_LISTNAME];

/* The Information Exchange message class */
char MsgClass[EXP_MAXLEN_RV_MSG_CLASS];
/* The translation table */
char XlateTbl[EXP_MAXLEN_RV_XLATE_TBL];

/* The message key */
char Msgkey[EXP_MAXLEN_RV_MSGKEY];

/* The start date */
char StartDate[EXP_MAXLEN_RV_START_DATE];

/* The end date */
char EndDate[EXP_MAXLEN_RV_END_DATE];

/* The time zone */
char GlTimeZone[EXP_MAXLEN_RV_GL_TIME_ZONE];

/* The nickname */
char Nickname[EXP_MAXLEN_RV_NICKNAME];

/* The address type */
char AddrType[EXP_MAXLEN_RV_ADDR_TYPE];

/* The option to receive data as EDI or unformatted */
char Receiveedi[EXP_MAXLEN_RV_RECEIVEEDI];
}
Input field definitions for ExpReplaceRecvOrdrReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpRecvOrdrReplacedRspStruct” on page 339.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**OrdrName**
The receive order name.

- **Max length:** EXP_MAXLEN_RV_ORDR_NAME (20)
- **Min length:** EXP_MINLEN_RV_ORDR_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the receive order must exist in the database.

**OrderDescr**
The description of the order.

- **Max length:** EXP_MAXLEN_RV_ORDER_DESCR (79)
- **Min length:** EXP_MINLEN_RV_ORDER_DESCR (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for printable data.
**IeSysid**
The Information Exchange system where the account ID and user ID are registered. If you specify IeSysid, you must specify IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_RV_IE_SYSID (3)
- Min length: EXP_MINLEN_RV_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- Max length: EXP_MAXLEN_RV_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_RV_IE_MBOX_ID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**TableType**
The alias-table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_RV_TABLE_TYPE (1)
- Min length: EXP_MINLEN_RV_TABLE_TYPE (1)
- Default value: Blank
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - G Alias table for global account.
  - O Alias table for a specific account.
  - P Alias table for private use.
### AliasTbl
The alias-table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Default value</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_RV_ALIAS_TBL (3)</td>
<td></td>
<td></td>
<td>Blank</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_ALIAS_TBL (1)</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Default value</td>
<td>Blank</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### Aliasname
This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Default value</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_RV_ALIASNAME (16)</td>
<td></td>
<td></td>
<td>Blank</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_ALIASNAME (1)</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Default value</td>
<td>Blank</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### Listname
The name of the list.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Default value</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_RV_LISTNAME (8)</td>
<td></td>
<td></td>
<td>Blank</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_RV_LISTNAME (1)</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Default value</td>
<td>Blank</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Input field definitions for ExpReplaceRecvOrdrReqStruct

**Nickname**
This is a nickname from the address book. Use Nickname with AddrType to determine the network address to use for sending or receiving data. If you specify Nickname, you must specify AddrType.

- **Max length:** EXP_MAXLEN_RV_NICKNAME (20)
- **Min length:** EXP_MINLEN_RV_NICKNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**AddrType**
The address type to use when sending or receiving data. Use AddrType with Nickname to determine the network address. If you specify AddrType, you must specify Nickname.

- **Max length:** EXP_MAXLEN_RV_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_RV_ADDR_TYPE (1)
- **Default value:** RM_DFLT_RV_ADDR_TYPE (E)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **A** Use the alias name and the alias table.
  - **D** Use a distribution list with the name specified in the Nickname field.
  - **E** Use the Information Exchange address.
  - **X** Use the Mail Exchange address.

**MsgClass**
The user-specified Information Exchange user message class.

- **Max length:** EXP_MAXLEN_RV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_RV_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
## Fileid
The file name where the received data is stored.

| Max length: | EXP_MAXLEN_RV_FILEID (65) |
| Min length: | EXP_MINLEN_RV_FILEID (1) |
| Required:   | Yes                       |
| Case sensitive: | No                       |
| Validation: | Yes, as a system file path. |

## Archiveid
The archival reference identifier the user wants Information Exchange to assign to the received files.

| Max length: | EXP_MAXLEN_RV_ARCHIVEID (8) |
| Min length: | EXP_MINLEN_RV_ARCHIVEID (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |

## Allfiles
The option to receive all files that match the RECEIVE specifications or just the first file in the mailbox that matches the RECEIVE specifications.

| Max length: | EXP_MAXLEN_RV_ALLFILES (1) |
| Min length: | EXP_MINLEN_RV_ALLFILES (1) |
| Default value: | RM_DFLT_RV_ALLFILES (Y) |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
|              | N  Receive first matching file. |
|              | Y  Receive all matching files. |
**Multifiles**
This determines whether the files the user receives are placed into separate files or are concatenated into one file.

- **Max length:** \text{EXP\_MAXLEN\_RV\_MULTFILES\_OPT\ (1)}
- **Min length:** \text{EXP\_MINLEN\_RV\_MULTFILES\_OPT\ (1)}
- **Default value:** \text{RM\_DFLT\_RV\_MULTFILES\_OPT\ (N)}
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - \text{N} Store the data in a single file.
  - \text{Y} Store the received files into individual files. Use the file masks specified to generate new file names, if necessary.

**Origfile**
This indicates whether the user wants to receive data into a file, using the original name from the sending system.

- **Max length:** \text{EXP\_MAXLEN\_RV\_ORIGFILE\ (1)}
- **Min length:** \text{EXP\_MINLEN\_RV\_ORIGFILE\ (1)}
- **Default value:** \text{RM\_DFLT\_RV\_ORIGFILE\ (N)}
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - \text{N} Do not use the original file name.
  - \text{Y} Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpReplaceRecvOrdReqStruct

**Format**
This indicates whether or not to receive data as a file or as e-mail.

- **Max length:** EXP_MAXLEN_RV_FORMAT (1)
- **Min length:** EXP_MINLEN_RV_FORMAT (1)
- **Default value:** RM_DFLT_RV_FORMAT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not format as e-mail.
  - **Y** Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.

**Recordsize**
This indicates whether the EDI data is broken up into fixed-length records by placing CRLF characters in the data at specified intervals.

- **Max length:** EXP_MAXLEN_RV_RECORDSIZE (3)
- **Min length:** EXP_MINLEN_RV_RECORDSIZE (1)
- **Max value:** EXP_MAXVAL_RV_RECORDSIZE (999)
- **Min value:** EXP_MINVAL_RV_RECORDSIZE (0)
- **Default value:** RM_DFLT_RV_RECORDSIZE (0)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, as numeric data.

**Processlen**
This controls the processing of length delimiters at the beginning of each record.

- **Max length:** EXP_MAXLEN_RV_PROCESSLEN (1)
- **Min length:** EXP_MINLEN_RV_PROCESSLEN (1)
- **Default value:** RM_DFLT_RV_PROCESSLEN (C)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **C** Convert the line-length delimiters to CRLF delimiters.
  - **I** Ignore line-length delimiters.
  - **R** Remove line-length delimiters.
**Autoedi**
This indicates whether EDI file processing is automatic if the CDH indicates EDI format.

- **Max length:** EXP_MAXLEN_RV_AUTOEDI (1)
- **Min length:** EXP_MINLEN_RV_AUTOEDI (1)
- **Default value:** RM_DFLT_RV_AUTOEDI (Y)
- **Required:** No
- **Case sensitive:** Yes

**Validation:**
- **Y** Automatically process the data as EDI.
- **N** Do not process the data as EDI.

**Ediopt**
This indicates whether CRLF characters are added after each segment delimiter in the EDI file.

- **Max length:** EXP_MAXLEN_RV_EDIOPT (1)
- **Min length:** EXP_MINLEN_RV_EDIOPT (1)
- **Default value:** RM_DFLT_RV_EDIOPT (N)
- **Required:** No
- **Case sensitive:** Yes

**Validation:**
- **N** Do not add CRLF at the end of EDI segments.
- **Y** Add CRLF characters at the end of EDI segments.
- **B** Add CRLF characters after number of bytes specified in Recordsize.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.
Requeued
This indicates whether to receive requeued files (files retrieved from archive into the mailbox).

- **Max length:** EXP_MAXLEN_RV_REQUEUED (1)
- **Min length:** EXP_MINLEN_RV_REQUEUED (1)
- **Default value:** RM_DFLT_RV_REQUEUED (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not receive requeued files.
  - **Y** Receive requeued files. Do not specify a destination on the receive command if you specify this option.

Removeeof
This indicates whether EOF characters are removed from the end of received files.

- **Max length:** EXP_MAXLEN_RV_REMOVEEOF (1)
- **Min length:** EXP_MINLEN_RV_REMOVEEOF (1)
- **Default value:** RM_DFLT_RV_REMOVEEOF (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not remove EOF characters.
  - **Y** Remove EOF characters.

Nonedionly
This specifies whether to receive only non-EDI-formatted data.

- **Max length:** EXP_MAXLEN_RV_NONEDIONLY (1)
- **Min length:** EXP_MINLEN_RV_NONEDIONLY (1)
- **Default value:** RM_DFLT_RV_NONEDIONLY (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Receive all data.
  - **Y** Check the CDH and receive only non-EDI data.
**Edionly**
This specifies whether or not to receive only EDI data.

**Max length:** EXP_MAXLEN_RV_EDIONLY (1)
**Min length:** EXP_MINLEN_RV_EDIONLY (1)
**Default value:** RM_DFLT_RV_EDIONLY (N)
**Required:** No
**Case sensitive:** Yes
**Validation:** The valid values are:
- N Receive all data.
- Y Check the CDH and receive only EDI data.

**Overwrite**
The action to take when receiving a file that already exists.

**Max length:** EXP_MAXLEN_RV_OVERWRITE (1)
**Min length:** EXP_MINLEN_RV_OVERWRITE (1)
**Default value:** RM_DFLT_RV_OVERWRITE (A)
**Required:** No
**Case sensitive:** Yes
**Validation:** The valid values are:
- A Append the new data to the existing file.
- G Generate a new file name using the file mask specified for the new data.
- O Overwrite the existing file.

**Waittime**
The number of minutes Expedite for Windows should wait to receive data.

**Max length:** EXP_MAXLEN_RV_WAITTIME (1)
**Min length:** EXP_MINLEN_RV_WAITTIME (1)
**Min value:** EXP_MAXVAL_RV_WAITTIME (0)
**Max value:** EXP_MINVAL_RV_WAITTIME (9)
**Default value:** RM_DFLT_RV_WAITTIME (0)
**Required:** No
**Validation:** Yes, as numeric data between 0 - 9.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpReplaceRecvOrdrReqStruct

**Msgkey**
The unique message key you use to receive a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_RV_MSGKEY (20)
- **Min length:** EXP_MINLEN_RV_MSGKEY (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**StartDate**
The starting date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_START_DATE (14)
- **Min length:** EXP_MINLEN_RV_START_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for date and time in YYYYMMDDHHMMSS format.

**EndDate**
The ending date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_END_DATE (14)
- **Min length:** EXP_MINLEN_RV_END_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for date and time in YYYYMMDDHHMMSS format.
GlTimeZone
Indicates whether the date and time field specified on the receive order is local time (as specified in the TimeZone field on the dropoff box or **ExpStartSess** function) or Greenwich mean time.

- Max length: EXP_MAXLEN_RV_GL_TIME_ZONE (1)
- Min length: EXP_MINLEN_RV_GL_TIME_ZONE (1)
- Default value: L
- Required: No
- Case sensitive: No
- Validation: The valid values are:
  - G Greenwich mean time (GMT).
  - L Local time.

XlateTbl
The translation table name.

- Max length: EXP_MAXLEN_RV_XLATE_TBL (8)
- Min length: EXP_MINLEN_RV_XLATE_TBL (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: Yes, it must contain the name of the translation table without the XLT extension.

Receiveedi
The option to receive data as EDI or unformatted.

- Max length: EXP_MAXLEN_RV_RECEIVEEDI (1)
- Min length: EXP_MINLEN_RV_RECEIVEEDI (1)
- Default value: N
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - N Receive as unformatted data.
  - Y Receive as EDI data.
Chapter 4. Writing the interface for the MailBox database

Output arguments for ExpReplaceRecvOrdr

```
struct ExpRecvOrdrReplacedRspStruct
{
  /* The order name for a receive */
  char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];

  /* The order description */
  char OrdrDescr[EXP_MAXLEN_RV_ORDR_DESCR];

  /* The Information Exchange system ID */
  char IeSysid[EXP_MAXLEN_RV_IE_SYSID];

  /* The Information Exchange account ID */
  char IeMboxId[EXP_MAXLEN_RV_IE_MBOX_ID];

  /* The Information Exchange table type; G, O, or P */
  char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

  /* The Information Exchange alias table */
  char AliasTbl[EXP_MAXLEN_RV_ALIAS_TBL];

  /* The Information Exchange alias name */
  char Aliasname[EXP_MAXLEN_RV_ALIASNAME];

  /* The list name */
  char Listname[EXP_MAXLEN_RV_LISTNAME];

  /* The nickname */
  char Nickname[EXP_MAXLEN_RV_NICKNAME];

  /* The address type */
  char AddrType[EXP_MAXLEN_RV_ADDR_TYPE];

  /* The Information Exchange message class */
  char MsgClass[EXP_MAXLEN_RV_MSG_CLASS];

  /* The file name for the received data */
  char Fileid[EXP_MAXLEN_RV_FILEID];

  /* The archive ID of received data */
  char Archiveid[EXP_MAXLEN_RV_ARCHIVEID];

  /* The option to receive one file or all files that match */
  char Allfiles[EXP_MAXLEN_RV_ALLFILES];

  /* The multiple files parameter for receive */
  char Multfiles[EXP_MAXLEN_RV_MULTFILES];

  /* The option to receive files with the orginal names */
  char Origfile[EXP_MAXLEN_RV_ORIGFILE];
}
```
/* The format for the data received */
char Format[EXP_MAXLEN_RV_FORMAT];

/* The EDI recordsize */
char Recordsize[EXP_MAXLEN_RV_RECORDSIZE];

/* The processing for line-length delimiters */
char Processlen[EXP_MAXLEN_RV_PROCESSLEN];

/* The automatic processing for EDI data */
char Autoedi[EXP_MAXLEN_RV_AUTOEDI];

/* The option for EDI data processing */
char Ediopt[EXP_MAXLEN_RV_EDIOPT];

/* Reserved field */
char Reserved1[20];

/* The requeued parameter for receive */
char Requeued[EXP_MAXLEN_RV_REQUEUED];

/* The option to remove EOF characters */
char Removeeof[EXP_MAXLEN_RV_REMOVEEOF];

/* The option to receive only non EDI data */
char Nonedionly[EXP_MAXLEN_RV_NONEDIONLY];

/* The option to receive only EDI data */
char Edionly[EXP_MAXLEN_RV_EDIONLY];

/* The option to overwrite, append, or generate a new filename */
char Overwrite[EXP_MAXLEN_RV_OVERWRITE];

/* The time to wait for data */
char Waittime[EXP_MAXLEN_RV_WAITTIME];

/* The message key */
char Msgkey[EXP_MAXLEN_RV_MSGKEY];

/* The start date */
char StartDate[EXP_MAXLEN_RV_START_DATE];

/* The end date */
char EndDate[EXP_MAXLEN_RV_END_DATE];

/* The time zone */
char GlTimeZone[EXP_MAXLEN_RV_GL_TIME_ZONE];

/* The translation table */
char XlateTbl[EXP_MAXLEN_RV_XLATE_TBL];

/* The option to receive as EDI or unformatted */
char Receiveedi[EXP_MAXLEN_RV_RECEIVEEDI];
};
Output field definitions for ExpRecvOrdrReplacedRspStruct

The output can include any or all of these fields.

**OrdrName**
The order name for a receive command.

- **Max length:** `EXP_MAXLEN_RV_ORDR_NAME` (20)
- **Min length:** `EXP_MINLEN_RV_ORDR_NAME` (1)
- **Contents:** Printable data.

**Fileid**
The file name where the received data is stored.

- **Max length:** `EXP_MAXLEN_RV_FILEID` (65)
- **Min length:** `EXP_MINLEN_RV_FILEID` (1)
- **Contents:** A system file path.

**Archiveid**
The archival reference identifier for Information Exchange to assign to the received files.

- **Max length:** `EXP_MAXLEN_RV_ARCHIVEID` (8)
- **Min length:** `EXP_MINLEN_RV_ARCHIVEID` (1)
- **Contents:** Printable data.

**Allfiles**
This indicates whether all the files that match the RECEIVE specifications are received or just the first file in the mailbox that matches the RECEIVE specifications.

- **Max length:** `EXP_MAXLEN_RV_ALLFILES` (1)
- **Min length:** `EXP_MINLEN_RV_ALLFILES` (1)
- **Contents:** The valid values are:
  - **N** Receive first matching file.
  - **Y** Receive all matching files.
Multfiles
This determines whether the files the user receives are placed into separate files or are concatenated into one file.

Max length: EXP_MAXLEN_RV_MULTFILES_OPT (1)
Min length: EXP_MINLEN_RV_MULTFILES_OPT (1)
Contents: One of these valid values:
N  Store the data in a single file.
Y  Store the received files into individual files. Use the file masks specified to generate new file names, if necessary.

Origfile
This indicates whether the user wants to receive data into a file, using the original name from the sending system.

Max length: EXP_MAXLEN_RV_ORIGFILE (1)
Min length: EXP_MINLEN_RV_ORIGFILE (1)
Contents: One of these valid values:
N  Do not use the original file name.
Y  Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.

Format
This indicates whether or not to receive data as a file or as e-mail.

Max length: EXP_MAXLEN_RV_FORMAT (1)
Min length: EXP_MINLEN_RV_FORMAT (1)
Contents: One of these valid values:
N  Do not format as e-mail.
Y  Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.
**Recordsize**
This indicates whether the EDI data is broken up into fixed-length records by placing CRLF characters in the data at specified intervals.

- **Max length:** EXP_MAXLEN_RV_RECORDSIZE (3)
- **Min length:** EXP_MINLEN_RV_RECORDSIZE (1)
- **Max value:** EXP_MAXVAL_RV_RECORDSIZE (999)
- **Min value:** EXP_MINVAL_RV_RECORDSIZE (0)
- **Contents:** Numeric data.

**Processlen**
This controls the processing of length delimiters at the beginning of each record.

- **Max length:** EXP_MAXLEN_RV_PROCESSLEN (1)
- **Min length:** EXP_MINLEN_RV_PROCESSLEN (1)
- **Contents:** One of these valid values:
  - C  Convert the line-length delimiters to CRLF delimiters.
  - I  Ignore line-length delimiters.
  - R  Remove line-length delimiters.

**Autoedi**
This indicates whether EDI file processing is automatic if the CDH indicates EDI format.

- **Max length:** EXP_MAXLEN_RV_AUTOEDI (1)
- **Min length:** EXP_MINLEN_RV_AUTOEDI (1)
- **Contents:** One of these valid values:
  - Y  Automatically process the data as EDI.
  - N  Do not process the data as EDI.
**Ediopt**
This indicates whether CRLF characters are added after each segment delimiter in the EDI file.

- **Max length:** EXP_MAXLEN_RV_EDIOPT (1)
- **Min length:** EXP_MINLEN_RV_EDIOPT (1)
- **Contents:** One of these valid values:
  - **N** Do not add CRLF at the end of EDI segments.
  - **Y** Add CRLF characters at the end of EDI segments.
  - **B** Add CRLF characters after number of bytes specified in Recordsize.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.

**Requeued**
This indicates whether to receive requeued files (files retrieved from archive into the mailbox).

- **Max length:** EXP_MAXLEN_RV_REQUEUED (1)
- **Min length:** EXP_MINLEN_RV_REQUEUED (1)
- **Contents:** One of these valid values:
  - **N** Do not receive requeued files.
  - **Y** Receive requeued files. Do not specify a destination on the receive command if you specify this option.

**Removeeof**
This indicates whether EOF characters are removed from the end of received files.

- **Max length:** EXP_MAXLEN_RV_REMOVEEOF (1)
- **Min length:** EXP_MINLEN_RV_REMOVEEOF (1)
- **Contents:** One of these valid values:
  - **N** Do not remove EOF characters.
  - **Y** Remove EOF characters.
Nonedionly
This specifies whether to receive only non-EDI data.

Max length: EXP_MAXLEN_RV_NONEDIONLY (1)
Min length: EXP_MINLEN_RV_NONEDIONLY (1)
Contents: One of these valid values:
N  Receive all data.
Y  Check the CDH and receive only non-EDI data.

Edionly
This specifies whether or not to receive only EDI data.

Max length: EXP_MAXLEN_RV_EDIONLY (1)
Min length: EXP_MINLEN_RV_EDIONLY (1)
Contents: One of these valid values:
N  Receive all data.
Y  Check the CDH and receive only EDI data.

Overwrite
The action to take when receiving a file that already exists.

Max length: EXP_MAXLEN_RV_OVERWRITE (1)
Min length: EXP_MINLEN_RV_OVERWRITE (1)
Contents: One of these valid values:
A  Append the new data to the existing file.
G  Generate a new file name using the file mask specified for the new data.
O  Overwrite the existing file.

Waittime
The number of minutes Expedite for Windows should wait to receive data.

Max length: EXP_MAXLEN_RV_WAITTIME (1)
Min length: EXP_MINLEN_RV_WAITTIME (1)
Min value: EXP_MAXVAL_RV_WAITTIME (0)
Max value: EXP_MINVAL_RV_WAITTIME (9)
Contents: Numeric data between 0 - 9.
**OrderDescr**
The description of the order.

- **Max length:** EXP_MAXLEN_RV_ORDER_DESCR (79)
- **Min length:** EXP_MINLEN_RV_ORDER_DESCR (1)
- **Contents:** Printable data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_RV_IE_SYSID (3)
- **Min length:** EXP_MINLEN_RV_IE_SYSID (1)
- **Contents:** Printable data.

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- **Max length:** EXP_MAXLEN_RV_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_RV_IE_MBOX_ID (1)
- **Contents:** Printable data.

**TableType**
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_RV_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_RV_TABLE_TYPE (1)
- **Contents:** One of these valid values:
  - **G** Alias table for global account.
  - **O** Alias table for a specific account.
  - **P** Alias table for private use.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvOrdrReplacedRspStruct

AliasTbl
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

- Max length: EXP_MAXLEN_RV_ALIAS_TBL (3)
- Min length: EXP_MINLEN_RV_ALIAS_TBL (1)
- Contents: Printable data.

Aliasname
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

- Max length: EXP_MAXLEN_RV_ALIASNAME (16)
- Min length: EXP_MINLEN_RV_ALIASNAME (1)
- Contents: Printable data.

Listname
The name of the list.

- Max length: EXP_MAXLEN_RV_LISTNAME (8)
- Min length: EXP_MINLEN_RV_LISTNAME (1)
- Contents: Printable data.

Nickname
The nickname from the address book. The Nickname is specified with AddrType to determine the network address for sending and receiving.

- Max length: EXP_MAXLEN_RV_NICKNAME (20)
- Min length: EXP_MINLEN_RV_NICKNAME (1)
- Contents: Printable data.
**AddrType**
The address type used for sending or receiving data. The AddrType is specified with Nickname to determine the network address.

- **Max length:** EXP_MAXLEN_RV_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_RV_ADDR_TYPE (1)
- **Contents:** One of these valid values:
  - A  Use the alias name and the alias table.
  - D  Use a distribution list with the name specified in the Nickname field.
  - E  Use the Information Exchange address.
  - X  Use the Mail Exchange address.

**MsgClass**
The Information Exchange user message class specified by the user.

- **Max length:** EXP_MAXLEN_RV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_RV_MSG_CLASS (1)
- **Contents:** Printable data.

**Msgkey**
The unique message key you use to receive a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_RV_MSGKEY (20)
- **Min length:** EXP_MINLEN_RV_MSGKEY (1)
- **Contents:** Printable data.

**StartDate**
The starting date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_START_DATE (14)
- **Min length:** EXP_MINLEN_RV_START_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**EndDate**
The ending date and time the user wants to receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_RV_END_DATE (14)
- **Min length:** EXP_MINLEN_RV_END_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.
**GlTimeZone**
Indicates whether the date and time field specified on the receive order is local time (as specified in the TimeZone field on the dropoff box or ExpStartSess function) or Greenwich mean time.

- **Max length:** `EXP_MAXLEN_RV_GL_TIME_ZONE (1)`
- **Min length:** `EXP_MINLEN_RV_GL_TIME_ZONE (1)`
- **Contents:** One of these valid values:
  - **G** Greenwich mean time (GMT).
  - **L** Local time.

**XlateTbl**
The translation table name.

- **Max length:** `EXP_MAXLEN_RV_XLATE_TBL (8)`
- **Min length:** `EXP_MINLEN_RV_XLATE_TBL (1)`
- **Contents:** Printable data.
ExpUsgRecvOrdrFromDropBox

This function unassigns the receive order from the dropoff box. If the receive order name or the dropoff box specified does not exist, the return code is set to 8. If the receive order is not assigned to the dropoff box, the return code is set to 4.

Function prototype ExpUsgRecvOrdrFromDropBox

```c
int ExpUsgRecvOrdrFromDropBox(
    struct ExpUsgRecvOrdrFromDropBoxReqStruct * Input
);
```

Input arguments for ExpUsgRecvOrdrFromDropBox

```c
struct ExpUsgRecvOrdrFromDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The receive order name */
    char OrdrName[EXP_MAXLEN_RV_ORDR_NAME];
};
```
Input field definitions for ExpUsgRecvOrdrFromDropBoxReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**OrdrName**
The order name for a receive command.

- Max length: EXP_MAXLEN_RV_ORDR_NAME (20)
- Min length: EXP_MINLEN_RV_ORDR_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the receive order must exist in the database.

**DropBoxName**
The dropoff box name.

- Max length: EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- Min length: EXP_MINLEN_DR_DROP_BOX_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the dropoff box must exist in the database.
Send orders
These are the functions, the input and output structures, and the field descriptions you use to work with send orders in the MailBox database.

ExpAddSendOrdr
This function adds a new send order to the MailBox database. If the send order name specified already exists, the return code is set to 8.

Function prototype for ExpAddSendOrdr

```c
int ExpAddSendOrdr(
    struct ExpAddSendOrdrReqStruct * Input,
    struct ExpSendOrdrAddedRspStruct * Output
);
```

Input arguments for ExpAddSendOrdr

```c
struct ExpAddSendOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The order name for sending */
    char OrdrName[EXP_MAXLEN_SN_ORDER_NAME];

    /* The description of the order */
    char OrderDescr[EXP_MAXLEN_SN_ORDER_DESCR];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_SN_IE_SYSID];

    /* The Information Exchange account ID and user ID*/
    char IeMboxID[EXP_MAXLEN_SN_IE_MBOX_ID];

    /* The alias-table type */
    char TableType[EXP_MAXLEN_SN_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_SN_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_SN_ALIASNAME];
}
```
Chapter 4. Writing the interface for the MailBox database

Input arguments for ExpAddSendOrdr

/* The list name */
char Listname[EXP_MAXLEN_SN_LISTNAME];

/* The nickname */
char Nickname[EXP_MAXLEN_SN_NICKNAME];

/* The address type */
char AddrType[EXP_MAXLEN_SN_ADDR_TYPE];

/* The Information Exchange message class */
char MsgClass[EXP_MAXLEN_SN_MSG_CLASS];

/* The name of the file to send */
char Fileid[EXP_MAXLEN_SN_FILEID];

/* The option to verify the receiver’s address */
char Verify[EXP_MAXLEN_SN_VERIFY];

/* The option to proceed if receiver verification fails */
char EdiVerifyProc[EXP_MAXLEN_SN_EDI_VERIFY_PROC];

/* The Information Exchange charging information */
char Charge[EXP_MAXLEN_SN_CHARGE];

/* The type of acknowledgment */
char Ack[EXP_MAXLEN_SN_ACK];

/* The retention period of the file */
char Retention[EXP_MAXLEN_SN_RETENTION];

/* The delivery service type for the file */
char Priority[EXP_MAXLEN_SN_PRIORITY];

/* Reserved field */
char Reserved1[20];

/* The type of data to send to Information Exchange */
char Datatype[EXP_MAXLEN_SN_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_SN_FORMAT];

/* The description of the file */
char Description[EXP_MAXLEN_SN_DESCRIPTION];

/* The status of data delimitation */
char Delimited[EXP_MAXLEN_SN_DELIMITED];

/* The record format of the file to be sent */
char Recformat[EXP_MAXLEN_SN_RECFORMAT];
/* The record length of the file */
char Recordlen[EXP_MAXLEN_SN_RECORDLEN];

/* The option to receive by msgclass, sender, or msgkey */
char Selectrcv[EXP_MAXLEN_SN_SELECTRCV];

/* The destination file name */
char Destfile[EXP_MAXLEN_SN_DESTFILE];

/* The destination file location */
char Destloc[EXP_MAXLEN_DESTLOC];

/* The indicator for test or normal files */
char Mode[EXP_MAXLEN_SN_MODE];

/* The message name */
char MsgName[EXP_MAXLEN_SN_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_SN_MSG_SEQ_NO];

/* The translation table */
char XlateTbl[EXP_MAXLEN_SN_XLATE_TBL];

/* The option to send as EDI or unformatted */
char Sendedi[EXP_MAXLEN_SENDEDI];
};
Input field definitions for ExpAddSendOrdrReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpSendOrdrAddedRspStruct” on page 369.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**OrdrName**
The order name for a send request.

- Max length: EXP_MAXLEN_SN_ORDR_NAME (20)
- Min length: EXP_MINLEN_SN_ORDR_NAME (1)
- Default value: Blank
- Required: Yes
- Case sensitive: No
- Validation: Yes, for printable data.

**OrderDescr**
The description of the send order.

- Max length: EXP_MAXLEN_SN_ORDER_DESCR (79)
- Min length: EXP_MINLEN_SN_ORDER_DESCR (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_SN_IE_SYSID (3)
- Min length: EXP_MINLEN_SN_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- Max length: EXP_MAXLEN_SN_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_SN_IE_MBOX_ID (9)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**TableType**
The alias-table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_SN_TABLE_TYPE (1)
- Min length: EXP_MINLEN_SN_TABLE_TYPE (1)
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - G Alias table for global account.
  - O Alias table for a specific account.
  - P Alias table for private use.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddSendOrdrReqStruct

**AliasTbl**
The alias-table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

- **Max length:** EXP_MAXLEN_SN_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_SN_ALIAS_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Aliasname**
This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_SN_ALIASNAME (16)
- **Min length:** EXP_MINLEN_SN_ALIASNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Listname**
The name of the list.

- **Max length:** EXP_MAXLEN_SN_LISTNAME (8)
- **Min length:** EXP_MINLEN_SN_LISTNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Input field definitions for ExpAddSendOrdrReqStruct

**Nickname**
This is a nickname from the address book. Use Nickname with AddrType to determine the network address to use for sending or receiving. If you specify Nickname, you must specify AddrType.

- **Max length:** EXP_MAXLEN_SN_NICKNAME (20)
- **Min length:** EXP_MINLEN_SN_NICKNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for printable data.

**AddrType**
The address type to use when sending or receiving. Use AddrType with Nickname to determine the network address. If you specify AddrType, you must specify Nickname.

- **Max length:** EXP_MAXLEN_SN_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_SN_ADDR_TYPE (1)
- **Default value:** RM_DFLT_SN_ADDR_TYPE (E)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **A** Use the alias name and the alias table.
  - **D** Use a distribution list with the name specified in the Nickname field.
  - **E** Use the Information Exchange address.
  - **X** Use the Mail Exchange address.

**MsgClass**
The user-specified Information Exchange user message class of the files to be sent. The receiver can use this name to receive only files of this particular class.

- **Max length:** EXP_MAXLEN_SN_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_SN_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Fileid**
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_SN_FILEID (54)
- **Min length:** EXP_MINLEN_SN_FILEID (1)
- **Default value:** Blank
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, as a file system path.

**Verify**
This indicates whether the verification of the receiver takes place before sending the file.

- **Max length:** EXP_MAXLEN_SN_VERIFY (1)
- **Min length:** EXP_MINLEN_SN_VERIFY (1)
- **Default value:** RM_DFLT_SN_VERIFY (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F: Verify that the receiver exists. If Information Exchange cannot verify, send the data anyway.
  - N: Do not verify the address.
  - Y: Verify the address. If the address is invalid or cannot be verified, do not send the data.

**EdiVerifyProc**
This option indicates what action to take if the address verification fails while sending envelopes from a file with multiple envelopes.

- **Max length:** EXP_MAXLEN_SNEDI_VERIFYPROC (1)
- **Min length:** EXP_MINLEN_SNEDI_VERIFYPROC (1)
- **Default value:** S
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - C: Continue processing.
  - S: Stop processing this file and continue with next command.
**Charge**

This indicates who pays the Information Exchange messages charges.

- **Max length:** EXP_MAXLEN_SN_CHARGE (1)
- **Min length:** EXP_MINLEN_SN_CHARGE (1)
- **Default value:** RM_DFLT_SN_CHARGE (3)
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  1. The receiver pays all charges.
  2. The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  3. The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  4. The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  5. The sender and receiver split all charges.
  6. The sender pays all charges.
**Ack**
The type of acknowledgment the user wants to receive. Information Exchange places acknowledgments in the user's mailbox.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_SN_ACK (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SN_ACK (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_SN_ACK (N)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td>A Information Exchange provides purge acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>B Information Exchange provides receipt and delivery acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>C Information Exchange provides receipt and purge acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>D Information Exchange provides delivery acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>E Information Exchange provides purge or delivery acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>F Information Exchange provides receipt and either purge or delivery acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>N Information Exchange does not provide acknowledgments.</td>
</tr>
<tr>
<td></td>
<td>R Information Exchange provides receipt acknowledgments.</td>
</tr>
</tbody>
</table>
**Retention**
The number of days Information Exchange keeps the file in the mailbox if no one receives it. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_SN_RETENTION (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SN_RETENTION (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_SN_RETENTION (0)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Max value:</td>
<td>EXP_MAXVAL_SN_RETENTION (180)</td>
</tr>
<tr>
<td>Min value:</td>
<td>EXP_MINVAL_SN_RETENTION (0)</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>No</td>
</tr>
<tr>
<td>Validation:</td>
<td>Yes, as numeric data.</td>
</tr>
</tbody>
</table>

**Priority**
The class of delivery service for the file to be sent.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_SN_PRIORITY (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_SN_PRIORITY (1)</td>
</tr>
<tr>
<td>Default value:</td>
<td>RM_DFLT_SN_PRIORITY (N)</td>
</tr>
<tr>
<td>Required:</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive:</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation:</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td><strong>N</strong> Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.</td>
</tr>
<tr>
<td></td>
<td><strong>H</strong> High priority. The file is placed at the beginning of the receiver’s queue.</td>
</tr>
<tr>
<td></td>
<td><strong>X</strong> Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.</td>
</tr>
</tbody>
</table>

**Reserved1**
This is a reserved field; leave it blank.
Length= 20.
Datatype
This indicates whether the data is sent in text or binary format. Binary data is not translated to EBCDIC, and is sent as is.

- **Max length:** \( \text{EXP\_MAXLEN\_SN\_DATATYPE} (1) \)
- **Min length:** \( \text{EXP\_MINLEN\_SN\_DATATYPE} (1) \)
- **Default value:** \( \text{RM\_DFLT\_SN\_DATATYPE} (T) \)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **B** Binary data.
  - **T** Text data.

Format
This indicates whether to send data as a file or as e-mail.

- **Max length:** \( \text{EXP\_MAXLEN\_SN\_FORMAT} (1) \)
- **Min length:** \( \text{EXP\_MINLEN\_SN\_FORMAT} (1) \)
- **Default value:** \( \text{RM\_DFLT\_SN\_FORMAT} (N) \)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Do not format as e-mail.
  - **Y** Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.

Description
This is a free-format file description.

- **Max length:** \( \text{EXP\_MAXLEN\_SN\_DESCRIPTION} (79) \)
- **Min length:** \( \text{EXP\_MINLEN\_SN\_DESCRIPTION} (1) \)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Delimited**
This indicates whether or not the data is delimited with CRLF characters.

- **Max length:** \texttt{EXP\_MAXLEN\_SN\_DELIMITED\_OPT} (1)
- **Min length:** \texttt{EXP\_MINLEN\_SN\_DELIMITED\_OPT} (1)
- **Default value:** \texttt{RM\_DFLT\_SN\_DELIMITED\_OPT} (Y)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Data is not delimited by CRLF characters.
  - Y Data is delimited with CRLF characters.

**Recformat**
The record format of the file. This information is stored in the CDH and does not affect processing on any system, except VM.

- **Max length:** \texttt{EXP\_MAXLEN\_SN\_RECFORMAT} (1)
- **Min length:** \texttt{EXP\_MINLEN\_SN\_RECFORMAT} (1)
- **Default value:** \texttt{RM\_DFLT\_SN\_RECFORMAT} (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F File has a fixed-record format.
  - N Record format is unknown.
  - V The file has a variable-record format.

**Recordlen**
The record length of the file to be sent. A value of 0 indicates the record length is unknown.

- **Max length:** \texttt{EXP\_MAXLEN\_SN\_RECORDLEN} (5)
- **Min length:** \texttt{EXP\_MINLEN\_SN\_RECORDLEN} (1)
- **Default value:** \texttt{RM\_DFLT\_SN\_RECORDLEN} (0)
- **Required:** No
- **Max value:** \texttt{EXP\_MAXVAL\_SN\_RECORDLEN} (65535)
- **Min value:** \texttt{EXP\_MINVAL\_SN\_RECORDLEN} (0)
- **Case sensitive:** No
- **Validation:** Yes, as numeric data.
Selectrcv
This indicates how the receiver can retrieve the file.

- **Max length:** EXP_MAXLEN_SN_SELECTRCV (1)
- **Min length:** EXP_MINLEN_SN_SELECTRCV (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  The receiver can retrieve the file as usual.
  - Y  The receiver must use the msgkey, the account ID, the user ID, or the user message class to receive this file.

Destfile
The original file name for the CDH. If you, as the receiver, specify the data should be received into a file with the same name as on the sender’s system, this file name is used if it is appropriate for your system. If you do not specify a destination file name, Expedite for Windows uses the file name on sender’s system for the CDH.

- **Max length:** EXP_MAXLEN_SN_DESTFILE (54)
- **Min length:** EXP_MINLEN_SN_DESTFILE (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

Destloc
The destination path for the CDH. This is provided for your information only. If you do not specify a destination path, Expedite for Windows uses the one from your system.

- **Max length:** EXP_MAXLEN_SN_DESTLOC (65)
- **Min length:** EXP_MINLEN_SN_DESTLOC (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Input field definitions for ExpAddSendOrdrReqStruct

**Mode**
This indicates a test file or a normal file.

- **Max length:** EXP_MAXLEN_SN_MODE (1)
- **Min length:** EXP_MINLEN_SN_MODE (1)
- **Default value:** RM_DFLT_SN_MODE (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Normal mode.
  - T Test mode.

**MsgName**
The user-specified message name.

- **Max length:** EXP_MAXLEN_SN_MSG_NAME (8)
- **Min length:** EXP_MINLEN_SN_MSG_NAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** EXP_MAXLEN_SN_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_SN_MSG_SEQ_NO (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpAddSendOrdrReqStruct

**XlateTbl**
The name of the translation table.

- **Max length:** EXP_MAXLEN_SN_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SN_XLATE_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, it must contain the file name of the translation table without the XLT extension.

**Sendedi**
The option to send data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_SN_SENDEDI (1)
- **Min length:** EXP_MINLEN_SN_SENDEDI (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Send as unformatted data.
  - Y  Send as EDI data.
Output arguments for ExpAddSendOrdr

```c
struct ExpSendOrdrAddedRspStruct {
    /* The order name for sending */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];

    /* The description of the order */
    char OrderDescr[EXP_MAXLEN_SN_ORDER_DESCR];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_SN_IE_SYSID];

    /* The Information Exchange account ID and user ID*/
    char IeMboxID[EXP_MAXLEN_SN_IE_MBOX_ID];

    /* The alias-table type */
    char TableType[EXP_MAXLEN_SN_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_SN_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_SN_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_SN_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_SN_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_SN_ADDR_TYPE];

    /* The Information Exchange message class */
    char MsgClass[EXP_MAXLEN_SN_MSG_CLASS];

    /* The name of the file to send */
    char Fileid[EXP_MAXLEN_SN_FILEID];

    /* The option to verify the receiver’s address */
    char Verify[EXP_MAXLEN_SN_VERIFY];

    /* The option to proceed if receiver verification fails */
    char EdiVerifyProc[EXP_MAXLEN_SN_EDI_VERIFY_PROC];

    /* The Information Exchange charging information */
    char Charge[EXP_MAXLEN_SN_CHARGE];
};
```
/* The type of acknowledgment */
char Ack[EXP_MAXLEN_SN_ACK];

/* The retention period of the file */
char Retention[EXP_MAXLEN_SN_RETENTION];

/* The delivery service type for the file */
char Priority[EXP_MAXLEN_SN_PRIORITY];

/* Reserved field */
char Reserved1[20];

/* The type of data to send to Information Exchange */
char Datatype[EXP_MAXLEN_SN_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_SN_FORMAT];

/* The description of the file */
char Description[EXP_MAXLEN_SN_DESCRIPTION];

/* The status of data delimitation */
char Delimited[EXP_MAXLEN_SN_DELIMITED];

/* The record format of the file to be sent */
char Recformat[EXP_MAXLEN_SN_RECFORMAT];

/* The record length of the file */
char Recordlen[EXP_MAXLEN_SN_RECORDLEN];

/* The option to receive by msgclass, sender, or msgkey */
char Selectrcv[EXP_MAXLEN_SN_SELECT_RCV];

/* The destination file name */
char Destfile[EXP_MAXLEN_SN_DESTFILE];

/* The destination file location */
char Destloc[EXP_MAXLEN_DESTLOC];

/* The indicator for test or normal files */
char Mode[EXP_MAXLEN_SN_MODE];

/* The message name */
char MsgName[EXP_MAXLEN_SN_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_SN_MSG_SEQ_NO];
/* The translation table */
char XlateTbl[EXP_MAXLEN_SN_XLATE_TBL];

/* The option to send as EDI or unformatted */
char Sendedi[EXP_MAXLEN_SENDEDI];
};
Output field definitions for ExpSendOrdrAddedRspStruct

The response can include any or all of these fields.

**OrdrName**
The order name for a send request.

- Max length: EXP_MAXLEN_SN_ORDR_NAME (20)
- Min length: EXP_MINLEN_SN_ORDR_NAME (1)
- Contents: Printable data.

**OrderDescr**
The description of the send order.

- Max length: EXP_MAXLEN_SN_ORDER_DESCR (79)
- Min length: EXP_MINLEN_SN_ORDER_DESCR (1)
- Contents: Printable data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_SN_IE_SYSID (3)
- Min length: EXP_MINLEN_SN_IE_SYSID (1)
- Contents: Printable data.

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- Max length: EXP_MAXLEN_SN_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_SN_IE_MBOX_ID (9)
- Contents: Printable data.
Output field definitions for ExpSendOrdrAddedRspStruct

**TableType**
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_SN_TABLE_TYPE (1)
- Min length: EXP_MINLEN_SN_TABLE_TYPE (1)
- Contents: One of these valid values:
  - G Alias table for global account.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

- Max length: EXP_MAXLEN_SN_ALIAS_TBL (3)
- Min length: EXP_MINLEN_SN_ALIAS_TBL (1)
- Contents: Printable data.

**Aliasname**
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

- Max length: EXP_MAXLEN_SN_ALIASNAME (16)
- Min length: EXP_MINLEN_SN_ALIASNAME (1)
- Contents: Printable data.

**Listname**
The name of the list.

- Max length: EXP_MAXLEN_SN_LISTNAME (8)
- Min length: EXP_MINLEN_SN_LISTNAME (1)
- Contents: Printable data.

**Nickname**
The nickname from the address book. The Nickname is specified with AddrType to determine the network address for sending or receiving.

- Max length: EXP_MAXLEN_SN_NICKNAME (20)
- Min length: EXP_MINLEN_SN_NICKNAME (1)
- Contents: Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrAddedRspStruct

AddrType
The address type used for sending or receiving. The AddrType is specified with Nickname to determine the network address. If you specify Addrtype, you must specify Nickname.

- **Max length**: EXP_MAXLEN_SN_ADDR_TYPE (1)
- **Min length**: EXP_MINLEN_SN_ADDR_TYPE (1)
- **Contents**: One of these valid values:
  - A Use the alias name and the alias table.
  - D Use a distribution list with the name specified in the Nickname field.
  - E Use the Information Exchange address.
  - X Use the Mail Exchange address.

MsgClass
The user-specified Information Exchange user message class of the files to be sent. The receiver can use this name to receive only files of this particular class.

- **Max length**: EXP_MAXLEN_SN_MSG_CLASS (8)
- **Min length**: EXP_MINLEN_SN_MSG_CLASS (1)
- **Contents**: Printable data.

Fileid
The file name where the received data is stored.

- **Max length**: EXP_MAXLEN_SN_FILEID (54)
- **Min length**: EXP_MINLEN_SN_FILEID (1)
- **Contents**: A file system path.

Verify
This indicates whether the verification of the receiver takes place before sending the file.

- **Max length**: EXP_MAXLEN_SN_VERIFY (1)
- **Min length**: EXP_MINLEN_SN_VERIFY (1)
- **Contents**: One of these valid values:
  - F Verify that the receiver exists. If Information Exchange cannot verify, send the data anyway.
  - N Do not verify the address.
  - Y Verify the address. If the address is invalid or cannot be verified, do not send the data.
**EdiVerifyProc**

This option indicates what action to take if the address verification fails while sending envelopes from a file with multiple envelopes.

- **Max length:** EXP_MAXLEN_SN_EDI_VERIFY_PROC (1)
- **Min length:** EXP_MINLEN_SN_EDI_VERIFY_PROC (1)
- **Contents:**
  - C Continue processing.
  - S Stop processing this file and continue with next command.

**Charge**

This indicates who pays the Information Exchange messages charges.

- **Max length:** EXP_MAXLEN_SN_CHARGE (1)
- **Min length:** EXP_MINLEN_SN_CHARGE (1)
- **Contents:**
  - 1 The receiver pays all charges.
  - 2 The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  - 3 The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  - 4 The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  - 5 The sender and receiver split all charges.
  - 6 The sender pays all charges.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrAddedRspStruct

**Ack**
The type of acknowledgment the user wants to receive. Information Exchange places acknowledgments in the user's mailbox.

| Max length: | EXP_MAXLEN_SN_ACK (1) |
| Min length: | EXP_MINLEN_SN_ACK (1) |
| Contents: | One of these valid values: |
| A | Information Exchange provides purge acknowledgments. |
| B | Information Exchange provides receipt and delivery acknowledgments. |
| C | Information Exchange provides receipt and purge acknowledgments. |
| D | Information Exchange provides delivery acknowledgments. |
| E | Information Exchange provides purge or delivery acknowledgments. |
| F | Information Exchange provides receipt and either purge or delivery acknowledgments. |
| N | Information Exchange does not provide acknowledgments. |
| R | Information Exchange provides receipt acknowledgments. |

**Retention**
The number of days Information Exchange keeps the file in the mailbox if no one receives it. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

| Max length: | EXP_MAXLEN_SN_RETENTION (3) |
| Min length: | EXP_MINLEN_SN_RETENTION (1) |
| Contents: | Numeric data. |
**Priority**
The class of delivery service for the file to be sent.

- **Max length:** EXP_MAXLEN_SN_PRIORITY (1)
- **Min length:** EXP_MINLEN_SN_PRIORITY (1)
- **Contents:** One of these valid values:
  - N Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.
  - H High priority. The file is placed at the beginning of the receiver’s queue.
  - X Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.

**Datatype**
This indicates whether the data is sent in text or binary format. Binary data is not translated to EBCDIC, and is sent as is.

- **Max length:** EXP_MAXLEN_SN_DATATYPE (1)
- **Min length:** EXP_MINLEN_SN_DATATYPE (1)
- **Contents:** One of these valid values:
  - B Binary data.
  - T Text data.

**Format**
This indicates whether to send data as a file or as e-mail.

- **Max length:** EXP_MAXLEN_SN_FORMAT (1)
- **Min length:** EXP_MINLEN_SN_FORMAT (1)
- **Contents:** One of these valid values.
  - N Do not format as e-mail.
  - Y Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrAddedRspStruct

Description
This is a free-format file description.

- **Max length:** EXP_MAXLEN_SN_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_SN_DESCRIPTION (1)
- **Contents:** Printable data.

Delimited
This indicates whether or not the data is delimited with CRLF characters.

- **Max length:** EXP_MAXLEN_SN_DELIMITED_OPT (1)
- **Min length:** EXP_MINLEN_SN_DELIMITED_OPT (1)
- **Contents:** One of these valid values:
  - **N** Data is not delimited by CRLF characters.
  - **Y** Data is delimited with CRLF characters.

Reformat
The record format of the file. This information is stored in the CDH and does not affect processing on any system, except VM.

- **Max length:** EXP_MAXLEN_SN_RECFORMAT (1)
- **Min length:** EXP_MINLEN_SN_RECFORMAT (1)
- **Contents:** One of these valid values:
  - **F** Fixed-record format.
  - **N** Record format is unknown.
  - **V** Variable-record format.

Recordlen
The record length of the file to be sent. A value of 0 indicates the record length is unknown.

- **Max length:** EXP_MAXLEN_SN_RECORDLEN (5)
- **Min length:** EXP_MINLEN_SN_RECORDLEN (1)
- **Contents:** Numeric data.
**Selectrcv**
This indicates how the receiver can retrieve the file.

- **Max length:** EXP_MAXLEN_SN_SELECTRCV (1)
- **Min length:** EXP_MINLEN_SN_SELECTRCV (1)
- **Contents:** One of these valid values:
  - N  The receiver can retrieve the file as usual.
  - Y  The receiver must use msgkey, account ID and user ID, or class to receive this file.

**Destfile**
The original file name for the CDH. If you, as the receiver, specify the data should be received into a file with the same name as on the sender’s system, this file name is used if it is appropriate for your system. If you do not specify a destination file name, Expedite for Windows uses the file name on your system for the CDH.

- **Max length:** EXP_MAXLEN_SN_DESTFILE (54)
- **Min length:** EXP_MINLEN_SN_DESTFILE (1)
- **Contents:** A file name.

**Destloc**
The destination path for the CDH. This is provided for the receiver’s information only. If you do not specify a destination path, Expedite for Windows uses the one from your system.

- **Max length:** EXP_MAXLEN_SN_DESTLOC (65)
- **Min length:** EXP_MINLEN_SN_DESTLOC (1)
- **Contents:** A file path.

**Mode**
This indicates a test or a normal file.

- **Max length:** EXP_MAXLEN_SN_MODE (1)
- **Min length:** EXP_MINLEN_SN_MODE (1)
- **Contents:** One of these valid values:
  - N  Normal mode.
  - T  Test mode.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrAddedRspStruct

**MsgName**
The user-specified message name.

- **Max length:** EXP_MAXLEN_SN_MSG_NAME (8)
- **Min length:** EXP_MINLEN_SN_MSG_NAME (1)
- **Contents:** Printable data.

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** EXP_MAXLEN_SN_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_SN_MSG_SEQ_NO (1)
- **Contents:** Printable data.

**XlateTbl**
The name of the translation table.

- **Max length:** EXP_MAXLEN_SN_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SN_XLATE_TBL (1)
- **Contents:** Printable data.

**Sendedi**
The option to send data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_SN_SENDEDI (1)
- **Min length:** EXP_MINLEN_SN_SENDEDI (1)
- **Contents:** One of these valid values:
  - N  Send as unformatted data.
  - Y  Send as EDI data.
ExpAsgSendOrdrToDropBox

This function assigns the send order to the dropoff box. If the send order name or dropoff box name specified does not exist, the return code is set to 8. If the send order name specified is already assigned to a dropoff box, the return code is set to 4.

Function prototype for ExpAsgSendOrdrToDropBox

```c
int ExpAsgSendOrdrToDropBox(
    struct ExpAsgSendOrdrToDropBoxReqStruct * Input
);
```

Input arguments for ExpAsgSendOrdrToDropBox

```c
struct ExpAsgSendOrdrToDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The send order name */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];
};
```
Input field definitions for ExpAsgSendOrdrToDropBoxReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

```
Max length:       EXP_MAXLEN_PJ_PROJ_NAME (20)
Min length:       EXP_MINLEN_PJ_PROJ_NAME (1)
Required:         Yes
Case sensitive:   No
Validation:       Yes, the project must be open.
```

**DropBoxName**
The dropoff box name.

```
Max length:       EXP_MAXLEN_DR_DROP_BOX_NAME (20)
Min length:       EXP_MINLEN_DR_DROP_BOX_NAME (1)
Required:         Yes
Case sensitive:   No
Validation:       Yes, the dropoff box must exist in the database.
```

**OrdrName**
The order name for a send request.

```
Max length:       EXP_MAXLEN_SN_ORDR_NAME (20)
Min length:       EXP_MINLEN_SN_ORDR_NAME (1)
Unique:           Yes, this is a key field.
Required:         Yes
Case sensitive:   No
Validation:       Yes, the send order must exist in the database.
```
ExpDeleteSendOrdr

This function deletes the send order from the MailBox database. If the send order name specified does not exist, the return code is set to 4.

If the send order is assigned to a dropoff box, it is unassigned before it is deleted. If this send order was the last one assigned to the dropoff box, the dropoff box remains in the MailBox database.

Function prototype for ExpDeleteSendOrdr

```c
int ExpDeleteSendOrdr(
    struct ExpDeleteSendOrdrReqStruct * Input
);
```

Input arguments for ExpDeleteSendOrdr

```c
struct ExpDeleteSendOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The order name for sending */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];
};
```
Input field definitions for ExpDeleteSendOrdrReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**OrdrName**
The order name for a send request.

- Max length: EXP_MAXLEN_SN_ORDR_NAME (20)
- Min length: EXP_MINLEN_SN_ORDR_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the send order must exist in the database.
ExpListSendOrdr

This function lists the send orders on the order shelf in the Mailbox database that correspond to the query formed in the request structure. If the query results in no send orders found, the return code is set to 4.

Function prototype for ExpListSendOrdr

```c
int ExpListSendOrdr(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpSendOrdrListedRspStruct *Output
);
```

Input arguments for ExpListSendOrdr:

```c
class ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListSendOrdrReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpSendOrdrListedRspStruct” on page 389.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Max length</th>
<th>Min length</th>
<th>Required</th>
<th>Case sensitive</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProjName</td>
<td>The project name.</td>
<td>EXP_MAXLEN_PJ_PROJ_NAME</td>
<td>EXP_MINLEN_PJ_PROJ_NAME</td>
<td>Yes</td>
<td>No</td>
<td>Yes, the project must be open.</td>
</tr>
<tr>
<td>SearchParm</td>
<td>The field on which to search</td>
<td>EXP_MAXLEN_CF_SEARCH_PAR</td>
<td>EXP_MINLEN_SEARCH_PAR</td>
<td>Yes, if a SearchValue is specified.</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>SearchOp</td>
<td>The operator to use for the search.</td>
<td>EXP_MAXLEN_CF_SEARCH_OPER</td>
<td>EXP_MINLEN_SEARCH_OPER</td>
<td>Yes, if a SearchParm is specified.</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**ProjName**

Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
Required: Yes
Case sensitive: No
Validation: Yes, the project must be open.

**SearchParm**

Max length: EXP_MAXLEN_CF_SEARCH_PARM (15)
Min length: EXP_MINLEN_SEARCH_PARM (1)
Required: Yes, if a SearchValue is specified.
Case sensitive: Yes
Validation: Yes, the search fields are:
- Name
- Description
- MailboxId
- OrdrName
- Class

**SearchOp**

Max length: EXP_MAXLEN_CF_SEARCH_OPER (2)
Min length: EXP_MINLEN_SEARCH_OPER (1)
Required: Yes, if a SearchParm is specified.
Case sensitive: No
Validation: Yes, the valid characters are: =, <, >, <=, or >=.
Input field definitions for ExpListSendOrdrReqStruct

**SearchValue**
The search value to use.

Max length: EXP_MAXLEN_CF_SEARCH_VALUE (25)
Min length: EXP_MINLEN_SEARCH_VALUE (1)
Required: Yes, if a SearchParm is specified.
Case sensitive: No
Validation: No

**SortParm**
The field on which to sort.

Max length: EXP_MAXLEN_CF_SORT_PARM (15)
Min length: EXP_MINLEN_SORT_PARM (1)
Required: Yes, if a SortOrder is specified.
Case sensitive: Yes
Validation: Yes, the sort fields are:
  • Name
  • Description
  • MailboxId
  • OrdrName
  • Class

**SortOrder**
The order in which to sort the list.

Max length: EXP_MAXLEN_CF_SORT_ORDER (1)
Min length: EXP_MINLEN_SORT_ORDER (1)
Required: No
Case sensitive: Yes
Validation: The valid values are:
  **RM_SORT_ASCENDING (A)**
  Sort the list in ascending order.
  **RM_SORT_DESCENDING (D)**
  Sort the list in descending order.
Chapter 4. Writing the interface for the MailBox database

Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListSendOrdrReqStruct

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Output arguments for ExpListSendOrdr

```c
struct ExpSendOrdrListedRspStruct
{
    /* The order type for sending */
    char OrdrType[EXP_MAXLEN_DR_ORDR_TYPE];

    /* The order name for sending */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];

    /* The order description */
    char OrderDescr[EXP_MAXLEN_SN_ORDER_DESCR];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_SN_IE_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_SN_IE_MBOX_ID];

    /* The Information Exchange alias-table type */
    char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_SN_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_SN_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_SN_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_SN_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_SN_ADDR_TYPE];

    /* The class of the file to be sent */
    char MsgClass[EXP_MAXLEN_SN_MSG_CLASS];

    /* The name of the file to send */
    char Fileid[EXP_MAXLEN_SN_FILEID];

    /* The option to verify the receiver's address */
    char Verify[EXP_MAXLEN_SN_VERIFY];

    /* The option to proceed if verification fails */
    char EdiVerifyProc[EXP_MAXLEN_SN_EDI_VERIFY_PROC];
};
```
/* The Information Exchange charging information */
char Charge[EXP_MAXLEN_SN_CHARGE];

/* The type of acknowledgment */
char Ack[EXP_MAXLEN_SN_ACK];

/* The retention period of the file */
char Retention[EXP_MAXLEN_SN_RETENTION];

/* The delivery service type for the file */
char Priority[EXP_MAXLEN_SN_PRIORITY];

/* Reserved field */
char Reserved1[20];

/* The type of data to send to Information Exchange */
char Datatype[EXP_MAXLEN_SN_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_SN_FORMAT];

/* The description of the file */
char Description[EXP_MAXLEN_SN_DESCRIPTION];

/* The status of data delimitation */
char Delimited[EXP_MAXLEN_SN_DELIMITED];

/* The record format of the file to be sent */
char Recformat[EXP_MAXLEN_SN_RECFORMAT];

/* The record length of the file */
char Recordlen[EXP_MAXLEN_SN_RECORDLEN];

/* The selective receive flag */
char Selectrcv[EXP_MAXLEN_SN_SELECTRCV];

/* The name of the file for the receiver */
char Destfile[EXP_MAXLEN_SN_DESTFILE];

/* The path where the file is located */
char Destloc[EXP_MAXLEN_SN_DESTLOC];

/* This indicates test or normal files */
char Mode[EXP_MAXLEN_SN_MODE];

/* The message name */
char MsgName[EXP_MAXLEN_SN_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_SN_MSG_SEQ_NO];
/* The translation table */
char XlateTbl[EXP_MAXLEN_SN_XLATE_TBL];

/* The option to send as EDI or as unformatted */
char Sendedi[EXP_MAXLEN_SN_SENDEDI];
};
Output field definitions for ExpSendOrdrListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of send orders copied into the output structure.

**OrdrType**
The order type for a send request.

Max length:  \( \text{EXP\_MAXLEN\_DR\_ORDR\_TYPE} \) (1)
Min length:  \( \text{EXP\_MINLEN\_DR\_ORDR\_TYPE} \) (1)
Contents:  Contains the character S.

**OrdrName**
The order name for a send request.

Max length:  \( \text{EXP\_MAXLEN\_SN\_ORDR\_NAME} \) (20)
Min length:  \( \text{EXP\_MINLEN\_SN\_ORDR\_NAME} \) (1)
Contents:  Printable data.

**OrderDescr**
The description of the send order.

Max length:  \( \text{EXP\_MAXLEN\_SN\_ORDER\_DESCR} \) (79)
Min length:  \( \text{EXP\_MINLEN\_SN\_ORDER\_DESCR} \) (1)
Contents:  Printable data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified along with the IeMboxId to create the Information Exchange address.

Max length:  \( \text{EXP\_MAXLEN\_SN\_IE\_SYSID} \) (3)
Min length:  \( \text{EXP\_MINLEN\_SN\_IE\_SYSID} \) (1)
Contents:  Printable data.
**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_SN_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_SN_IE_MBOX_ID (9)
- **Contents:** Printable data.

**TableType**
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_SN_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_SN_TABLE_TYPE (1)
- **Contents:** One of these valid values:
  - G  Alias table for global account.
  - O  Alias table for a specific account.
  - P  Alias table for private use.

**AliasTbl**
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with the TableType and Aliasname.

- **Max length:** EXP_MAXLEN_SN_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_SN_ALIAS_TBL (1)
- **Contents:** Printable data.

**Aliasname**
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_SN_ALIASNAME (16)
- **Min length:** EXP_MINLEN_SN_ALIASNAME (1)
- **Contents:** Printable data.

**Listname**
The name of the list.

- **Max length:** EXP_MAXLEN_SN_LISTNAME (8)
- **Min length:** EXP_MINLEN_SN_LISTNAME (1)
- **Contents:** Printable data.
Output field definitions for ExpSendOrdrListedRspStruct

Nickname
The nickname from the address book. The Nickname is specified with AddrType to determine the network address to use for sending or receiving.

Max length: EXP_MAXLEN_SN_NICKNAME (20)
Min length: EXP_MINLEN_SN_NICKNAME (1)
Contents: Printable data.

AddrType
The address type used for sending or receiving. The AddrType is specified with Nickname to determine the network address.

Max length: EXP_MAXLEN_SN_ADDR_TYPE (1)
Min length: EXP_MINLEN_SN_ADDR_TYPE (1)
Contents: One of these valid values:
A  Use the alias name and alias table.
D  Use a distribution list with the name specified in the Nickname field.
E  Use the Information Exchange address.
X  Use the Mail Exchange address.

MsgClass
The user-specified Information Exchange user message class of the files to be sent. The receiver can use this name to receive only files of this particular class.

Max length: EXP_MAXLEN_SN_MSG_CLASS (8)
Min length: EXP_MINLEN_SN_MSG_CLASS (1)
Contents: Printable data.

Fileid
The file name where the received data is stored.

Max length: EXP_MAXLEN_SN_FILEID (65)
Min length: EXP_MINLEN_SN_FILEID (1)
Contents: A file system path.
Verify
This indicates whether the verification of the receiver takes place before sending the file.

Max length: EXP_MAXLEN_SN_VERIFY (1)
Min length: EXP_MINLEN_SN_VERIFY (1)
Contents: One of these valid values:
   F  Verify that the receiver exists. If Information Exchange cannot verify, send the data anyway.
   N  Do not verify the address.
   Y  Verify the address. If the address is invalid or cannot be verified, do not send the data.

EdiVerifyProc
The option to proceed if the address verification fails while sending envelopes from a file with multiple envelopes.

Max length: EXP_MAXLEN_SN EDI_VERIFY_PROC (1)
Min length: EXP_MINLEN_SN EDI_VERIFY_PROC (1)
Contents: One of these valid values:
   C  Continue processing.
   S  Stop processing this file and continue with next command.
Charge
This indicates who pays the Information Exchange messages charges.

Max length: EXP_MAXLEN_SN_CHARGE (1)
Min length: EXP_MINLEN_SN_CHARGE (1)
Contents: One of these valid values:

1  The receiver pays all charges.
2  The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
3  The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
4  The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
5  The sender and receiver split all charges.
6  The sender pays all charges.
Ack
The type of acknowledgment the user wants to receive. Information Exchange places
acknowledgments in the user's mailbox.

Max length: EXP_MAXLEN_SN_ACK (1)
Min length: EXP_MINLEN_SN_ACK (1)
Contents: One of these valid values:
A  Information Exchange provides purge acknowledgments.
B  Information Exchange provides receipt and delivery acknowledgments.
C  Information Exchange provides receipt and purge acknowledgments.
D  Information Exchange provides delivery acknowledgments.
E  Information Exchange provides purge or delivery acknowledgments.
F  Information Exchange provides receipt and either purge or delivery acknowledgments.
N  Information Exchange does not provide acknowledgments.
R  Information Exchange provides receipt acknowledgments.

Retention
The number of days Information Exchange keeps the file in the mailbox if no one
receives it. The default value of 0 indicates that Information Exchange should use the
system default, which varies with the Information Exchange installation. Currently,
the default in the U.S. is 30 days.

Max length: EXP_MAXLEN_SN_RETENTION (3)
Min length: EXP_MINLEN_SN_RETENTION (1)
Contents: Numeric data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrListedRspStruct

**Priority**
The class of delivery service for the file to be sent.

- **Max length:** EXP_MAXLEN_SN_PRIORITY (1)
- **Min length:** EXP_MINLEN_SN_PRIORITY (1)
- **Contents:** One of these valid values:
  - **N** Normal priority. The file is received in the order placed in the mailbox. Normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.

**Datatype**
This indicates whether the data is sent in text or binary format. Binary data is not translated to EBCDIC and is sent as is.

- **Max length:** EXP_MAXLEN_SN_DATATYPE (1)
- **Min length:** EXP_MINLEN_SN_DATATYPE (1)
- **Contents:** One of these valid values:
  - **B** Binary data.
  - **T** Text data.

**Format**
This indicates whether to send data as a file or as e-mail.

- **Max length:** EXP_MAXLEN_SN_FORMAT (1)
- **Min length:** EXP_MINLEN_SN_FORMAT (1)
- **Contents:** One of these valid values:
  - **N** Do not format as e-mail.
  - **Y** Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.
Description
This is a free-format file description.

Max length: EXP_MAXLEN_SN_DESCRIPTION (79)
Min length: EXP_MINLEN_SN_DESCRIPTION (1)
Contents: Printable data.

Delimited
This indicates whether or not the data is delimited with CRLF characters.

Max length: EXP_MAXLEN_SN_DELIMITED_OPT (1)
Min length: EXP_MINLEN_SN_DELIMITED_OPT (1)
Contents: One of these valid values:
N  Data is not delimited by CRLF characters.
Y  Data is delimited with CRLF characters.

Reccformat
The record format of the file. This information is stored in the CDH and does not affect processing on any system, except VM.

Max length: EXP_MAXLEN_SN_RECFORMAT (1)
Min length: EXP_MINLEN_SN_RECFORMAT (1)
Contents: One of these valid values:
F  Fixed-record format.
N  Record format is unknown.
V  Variable-record format.

Recordlen
The record length of the file to be sent. A value of 0 indicates the record length is unknown.

Max length: EXP_MAXLEN_SN_RECORDLEN (5)
Min length: EXP_MINLEN_SN_RECORDLEN (1)
Contents: Numeric data.
Selectrcv
This indicates how the receiver can retrieve the file.

Max length: EXP_MAXLEN_SN_SELECTRCV (1)
Min length: EXP_MINLEN_SN_SELECTRCV (1)
Contents: One of these valid values:
  N  The receiver can retrieve the file as usual.
  Y  The receiver must use msgkey, account ID and user ID, or class to receive this file.

Destfile
The original file name for the CDH. If you, as the receiver, specify the data should be received into a file with the same name as on the sender’s system, this file name is used if it is appropriate for your system. If you do not specify a destination file name, Expedite for Windows uses the file name on your system for the CDH.

Max length: EXP_MAXLEN_SN_DESTFILE (54)
Min length: EXP_MINLEN_SN_DESTFILE (1)
Contents: A file name.

Destloc
The destination path for the CDH. This is provided for the receiver’s information only. If you do not specify a destination path, Expedite for Windows uses the one from your system.

Max length: EXP_MAXLEN_SN_DESTLOC (65)
Min length: EXP_MINLEN_SN_DESTLOC (1)
Contents: A file path.

Mode
This indicates a test or a normal file.

Max length: EXP_MAXLEN_SN_MODE (1)
Min length: EXP_MINLEN_SN_MODE (1)
Contents: One of these valid values:
  N  Normal mode.
  T  Test mode.
Output field definitions for ExpSendOrdrListedRspStruct

**MsgName**
The user-specified message name.

- **Max length:** EXP_MAXLEN_SN_MSG_NAME (8)
- **Min length:** EXP_MINLEN_SN_MSG_NAME (1)
- **Contents:** Printable data.

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** EXP_MAXLEN_SN_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_SN_MSG_SEQ_NO (1)
- **Contents:** Printable data.

**XlateTbl**
The name of the translation table.

- **Max length:** EXP_MAXLEN_SN_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SN_XLATE_TBL (1)
- **Contents:** Printable data.

**Sendedi**
The option to send data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_SN_SENDEDI (1)
- **Min length:** EXP_MINLEN_SN_SENDEDI (1)
- **Contents:** One of these valid values:
  - N  Send as unformatted data.
  - Y  Send as EDI data.
ExpReplaceSendOrdr

This function replaces information about one of the send orders. If the send order name specified does not exist, the return code is set to 8.

Function prototype ExpReplaceSendOrdr

```c
int ExpReplaceSendOrdr(
    struct ExpReplaceSendOrdrReqStruct * Input,
    struct ExpSendOrdrReplacedRspStruct * Output
);
```

Input arguments for ExpReplaceSendOrdr

```c
struct ExpReplaceSendOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The send order name */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];

    /* The name of the file to send */
    char Fileid[EXP_MAXLEN_SN_FILEID];

    /* The option to verify receiver’s address */
    char Verify[EXP_MAXLEN_SN_VERIFY];

    /* The option to stop or continue when an error occurs */
    char EdiVerifyProc[EXP_MAXLEN_SN_EDI_VERIFY_PROC];

    /* The Information Exchange charging information */
    char Charge[EXP_MAXLEN_SN_CHARGE];

    /* The type of acknowledgement */
    char Ack[EXP_MAXLEN_SN_ACK];

    /* The retention period of the file */
    char Retention[EXP_MAXLEN_SN_RETENTION];

    /* The delivery service type for the file */
    char Priority[EXP_MAXLEN_SN_PRIORITY];

    /* Reserved field */
    char Reserved1[20];
```
/* The type of data to send to Information Exchange */
char Datatype[EXP_MAXLEN_SN_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_SN_FORMAT];

/* The description of the file */
char Description[EXP_MAXLEN_SN_DESCRIPTION];

/* The status of data delimitation */
char Delimited[EXP_MAXLEN_SN_DELIMITED];

/* The record format of the file to be sent */
char Recformat[EXP_MAXLEN_SN_RECFORMAT];

/* The record length of the file */
char Recordlen[EXP_MAXLEN_SN_RECORDLEN];

/* The option to prevent blanket receive */
char Selectrcv[EXP_MAXLEN_SN_SELECTRCV];

/* The destination file name */
char Destfile[EXP_MAXLEN_SN_DESTFILE];

/* The destination file location */
char Destloc[EXP_MAXLEN_SN_DESTLOC];

/* This indicates test or normal files */
char Mode[EXP_MAXLEN_SN_MODE];

/* The description of the order */
char OrdrDescr[EXP_MAXLEN_SN_ORDR_DESCR];

/* The Information Exchange system ID */
char IeSysid[EXP_MAXLEN_SN_IE_SYSID];

/* The Information Exchange account ID */
char IeMboxId[EXP_MAXLEN_SN_IE_MBOX_ID];

/* The Information Exchange table type, G, O, or P */
char TableType[EXP_MAXLEN_SN_TABLE_TYPE];

/* The Information Exchange alias table */
char AliasTbl[EXP_MAXLEN_SN_ALIAS_TBL];

/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_SN_ALIASNAME];

/* The list name */
char Listname[EXP_MAXLEN_SN_LISTNAME];
/* The nickname */
char Nickname[EXP_MAXLEN_SN_NICKNAME];

/* The address type */
char AddrType[EXP_MAXLEN_SN_ADDR_TYPE];

/* The Information Exchange message class */
char MsgClass[EXP_MAXLEN_SN_MSG_CLASS];

/* The message name */
char MsgName[EXP_MAXLEN_SN_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_SN_MSG_SEQ_NO];

/* The translation table */
char XlateTbl[EXP_MAXLEN_SN_XLATE_TBL];

/* The option to send data as EDI or unformatted */
char Sendedi[EXP_MAXLEN_SN_SENDEDI];
};
**Input field definitions for ExpReplaceSendOrdrReqStruct**

For a list of output field descriptions, see “Output field definitions for ExpSendOrdrReplacedRspStruct” on page 417.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

| Max length: | EXP_MAXLEN_PJ_PROJ_NAME (20) |
| Min length: | EXP_MINLEN_PJ_PROJ_NAME (20) |
| Required:   | Yes                        |
| Case sensitive: | No                        |
| Validation: | Yes, the project must be open. |

**OrdrName**
The order name for a send request.

| Max length: | EXP_MAXLEN_SN_ORDR_NAME (20) |
| Min length: | EXP_MINLEN_SN_ORDR_NAME (1)  |
| Default value: | Blank                      |
| Required:   | Yes                        |
| Case sensitive: | No                        |
| Validation: | Yes, the order must exist in the database. |

**OrderDescr**
The description of the send order.

| Max length: | EXP_MAXLEN_SN_ORDER_DESCR (79) |
| Min length: | EXP_MINLEN_SN_ORDER_DESCR (1)  |
| Default value: | Blank                      |
| Required:   | No                          |
| Case sensitive: | No                        |
| Validation: | Yes, for printable data. |
**IeSysid**
The Information Exchange system where the account ID and user ID are registered. If you specify the IeSysid, you must specify the IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_SN_IE_SYSID (3)
- Min length: EXP_MINLEN_SN_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- Max length: EXP_MAXLEN_SN_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_SN_IE_MBOX_ID (9)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**TableType**
The alias-table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_DV_TABLE_TYPE (1)
- Min length: EXP_MINLEN_DV_TABLE_TYPE (1)
- Default value: Blank
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - **G** Alias table for global account.
  - **O** Alias table for a specific account.
  - **P** Alias table for private use.
AliasTbl
The alias-table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

- Max length: EXP_MAXLEN_SN_ALIAS_TBL (3)
- Min length: EXP_MINLEN_SN_ALIAS_TBL (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

Aliasname
This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

- Max length: EXP_MAXLEN_SN_ALIASNAME (16)
- Min length: EXP_MINLEN_SN_ALIASNAME (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

Listname
The name of the list.

- Max length: EXP_MAXLEN_SN_LISTNAME (8)
- Min length: EXP_MINLEN_SN_LISTNAME (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpReplaceSendOrdrReqStruct

Nickname
This is a nickname from the address book. Use Nickname with AddrType to determine the network address to use for sending or receiving. If you specify Nickname, you must specify AddrType.

Max length: EXP_MAXLEN_SN_NICKNAME (20)
Min length: EXP_MINLEN_SN_NICKNAME (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: Yes, for printable data.

AddrType
The address type to use when sending or receiving. Use AddrType with Nickname to determine the network address. If you specify AddrType, you must specify Nickname.

Max length: EXP_MAXLEN_SN_ADDR_TYPE (1)
Min length: EXP_MINLEN_SN_ADDR_TYPE (1)
Default value: RM_DFLT_SN_ADDR_TYPE (E)
Required: No
Case sensitive: Yes
Validation: The valid values are:
A Use the alias name and the alias table.
D Use a distribution list with the name specified in the Nickname field.
E Use the Information Exchange address.
X Use the Mail Exchange address.
**MsgClass**

The user-specified Information Exchange user message class of the files to be sent. The receiver can use this name to receive only files of this particular class.

- **Max length:** `EXP_MAXLEN_SN.MSG_CLASS` (8)
- **Min length:** `EXP_MINLEN_SN.MSG_CLASS` (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Fileid**

The file name where the received data is stored.

- **Max length:** `EXP_MAXLEN_SN.FILEID` (54)
- **Min length:** `EXP_MINLEN_SN.FILEID` (1)
- **Default value:** Blank
- **Required:** Yes
- **Case sensitive:** Yes
- **Validation:** Yes, as a file system path.

**Verify**

This indicates whether the verification of the receiver takes place before sending the file.

- **Max length:** `EXP_MAXLEN_SNVERIFY` (1)
- **Min length:** `EXP_MINLEN_SNVERIFY` (1)
- **Default value:** `RM_DFLT_SN_VERIFY` (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **F** Verify that the receiver exists. If Information Exchange cannot verify, send the data anyway.
  - **N** Do not verify the address.
  - **Y** Verify the address. If the address is invalid or cannot be verified, do not send the data.
**EdiVerifyProc**
This option to proceed if address verification fails while sending envelopes from a file with multiple envelopes.

Max length: EXP_MAXLEN_SN_EDI_VERIFY_PROC (1)
Min length: EXP_MINLEN_SN_EDI_VERIFY_PROC (1)
Default value: S
Required: No
Case sensitive: Yes
Validation: The valid values are:
C Continue processing.
S Stop processing this file and continue with next command.

**Charge**
This indicates who pays the Information Exchange messages charges.

Max length: EXP_MAXLEN_SN_CHARGE (1)
Min length: EXP_MINLEN_SN_CHARGE (1)
Default value: RM_DFLT_SN_CHARGE (3)
Required: No
Case sensitive: No
Validation: The valid values are:
1 The receiver pays all charges.
2 The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
3 The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
4 The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
5 The sender and receiver split all charges.
6 The sender pays all charges.
Ack
The type of acknowledgment the user wants to receive. Information Exchange places acknowledgments in the user's mailbox.

Max length: EXP_MAXLEN_SN_ACK (1)
Min length: EXP_MINLEN_SN_ACK (1)
Default value: RM_DFLT_SN_ACK (N)
Required: No
Case sensitive: Yes
Validation: The valid values are:

A Information Exchange provides purge acknowledgments.
B Information Exchange provides receipt and delivery acknowledgments.
C Information Exchange provides receipt and purge acknowledgments.
D Information Exchange provides delivery acknowledgments.
E Information Exchange provides purge or delivery acknowledgments.
F Information Exchange provides receipt and either purge or delivery acknowledgments.
N Information Exchange does not provide acknowledgments.
R Information Exchange provides receipt acknowledgments.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpReplaceSendOrdReqStruct

Retention
The number of days Information Exchange keeps the file in the mailbox, if no one receives it. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- Max length: EXP_MAXLEN_SN_RETENTION (3)
- Min length: EXP_MINLEN_SN_RETENTION (1)
- Max value: EXP_MAXVAL_SN_RETENTION (180)
- Min value: EXP_MINVAL_SN_RETENTION (0)
- Default value: RM_DFLT_SN_RETENTION (0)
- Required: No
- Case sensitive: No
- Validation: Numeric data.

Priority
The class of delivery service for the file to be sent.

- Max length: EXP_MAXLEN_SN_PRIORITY (1)
- Min length: EXP_MINLEN_SN_PRIORITY (1)
- Default value: RM_DFLT_SN_PRIORITY (N)
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - N Normal priority. The file is received in the order placed in the mailbox. Normal priority is first-in, first-out.
  - H High priority. The file is placed at the beginning of the receiver’s queue.
  - X Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

Reserved1
This is a reserved field; leave it blank.
Length = 20.
### Datatype
This indicates whether the data is sent in text or binary format. Binary data is not translated to EBCDIC and is sent as is.

| Max length: | EXP_MAXLEN_SN_DATATYPE (1) |
| Min length: | EXP_MINLEN_SN_DATATYPE (1) |
| Default value: | RM_DFLT_SN_DATATYPE (T) |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
| | B  Binary data. |
| | T  Text data. |

### Format
This indicates whether to send data as a file or as e-mail.

| Max length: | EXP_MAXLEN_SN_FORMAT (1) |
| Min length: | EXP_MINLEN_SN_FORMAT (1) |
| Default value: | RM_DFLT_SN_FORMAT (N) |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
| | N  Do not format as e-mail. |
| | Y  Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001. |

### Description
This is a free-format file description.

| Max length: | EXP_MAXLEN_SN_DESCRIPTION (79) |
| Min length: | EXP_MINLEN_SN_DESCRIPTION (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpReplaceSendOrdReqStruc

Delimited
This indicates whether or not the data is delimited with CRLF characters.

- **Max length:** EXP_MAXLEN_SN_DELIMITED_OPT (1)
- **Min length:** EXP_MINLEN_SN_DELIMITED_OPT (1)
- **Default value:** RM_DFLT_SN_DELIMITED_OPT (Y)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Data is not delimited by CRLF characters.
  - Y Data is delimited with CRLF characters.

Recformat
The record format of the file. This information is stored in the CDH and does not affect processing on any system, except VM.

- **Max length:** EXP_MAXLEN_SN_RECFORMAT (1)
- **Min length:** EXP_MINLEN_SN_RECFORMAT (1)
- **Default value:** RM_DFLT_SN_RECFORMAT (N)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F Fixed-record format.
  - N Record format is unknown.
  - V Variable-record format.

Recordlen
The record length of the file. A value of 0 indicates the record length is unknown.

- **Max length:** EXP_MAXLEN_SN_RECORDLEN (5)
- **Min length:** EXP_MINLEN_SN_RECORDLEN (1)
- **Max value:** EXP_MAXVAL_SN_RECORDLEN (65535)
- **Min value:** EXP_MINVAL_SN_RECORDLEN (0)
- **Default value:** RM_DFLT_SN_RECORDLEN (0)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, as numeric data.
Selectrcv
This indicates how the receiver can retrieve the file.

| Max length: | EXP_MAXLEN_SN_SELECTRCV (1) |
| Min length: | EXP_MINLEN_SN_SELECTRCV (1) |
| Default value: | N |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: N The receiver can retrieve the file as usual. Y The receiver must use msgkey, account ID and user ID, or class to receive this file. |

Destfile
The original file name for the CDH. If you, as the receiver, specify the data should be received into a file with the same name as on the sender’s system, this file name is used if it is appropriate for your system. If you do not specify a destination file name, Expedite for Windows uses the file name on your system for the CDH.

| Max length: | EXP_MAXLEN_SN_DESTFILE (54) |
| Min length: | EXP_MINLEN_SN_DESTFILE (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |

Destloc
The destination path for the CDH. This is provided for the receiver’s information only. If you do not specify a destination path, Expedite for Windows uses the one from your system.

| Max length: | EXP_MAXLEN_SN_DESTLOC (65) |
| Min length: | EXP_MINLEN_SN_DESTLOC (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |
**Mode**
This indicates a test or a normal file.

- **Max length:** `EXP_MAXLEN_SN_MODE (1)`
- **Min length:** `EXP_MINLEN_SN_MODE (1)`
- **Default value:** `RM_DFLT_SN_MODE (N)`
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Normal mode.
  - T  Test mode.

**MsgName**
The user-specified message name.

- **Max length:** `EXP_MAXLEN_SN_MSG_NAME (8)`
- **Min length:** `EXP_MINLEN_SN_MSG_NAME (1)`
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** `EXP_MAXLEN_SN_MSG_SEQ_NO (5)`
- **Min length:** `EXP_MINLEN_SN_MSG_SEQ_NO (1)`
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**XlateTbl**

The name of the translation table.

- **Max length:** EXP_MAXLEN_SN_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SN_XLATE_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, it must contain the file name of the translation table without the XLT extension.

**Sendedi**

The option to send data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_SN_SENDEDI (1)
- **Min length:** EXP_MINLEN_SN_SENDEDI (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Send as unformatted data.
  - Y  Send as EDI data.
Output arguments for ExpReplaceSendOrdr

**struct ExpSendOrdrReplacedRspStruct**

{  
/* The send order name */  
char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];  

/* The send order description */  
char OrderDescr[EXP_MAXLEN_SN_ORDER_DESCR];  

/* The Information Exchange system ID */  
char IeSysid[EXP_MAXLEN_SN_IE_SYSID];  

/* The Information Exchange account ID and user ID */  
char IeMboxID[EXP_MAXLEN_SN_IE_MBOX_ID];  

/* The alias-table type */  
char TableType[EXP_MAXLEN_SN_TABLE_TYPE];  

/* The Information Exchange alias table */  
char AliasTbl[EXP_MAXLEN_SN_ALIAS_TBL];  

/* The Information Exchange alias name */  
char Aliasname[EXP_MAXLEN_SN_ALIASNAME];  

/* The list name */  
char Listname[EXP_MAXLEN_SN_LISTNAME];  

/* The nickname */  
char Nickname[EXP_MAXLEN_SN_NICKNAME];  

/* The address type */  
char AddrType[EXP_MAXLEN_SN_ADDR_TYPE];  

/* The Information Exchange message class */  
char MsgClass[EXP_MAXLEN_SN_MSG_CLASS];  

/* The name of the file to send */  
char Fileid[EXP_MAXLEN_SN_FILEID];  

/* The option to verify receiver’s address */  
char Verify[EXP_MAXLEN_SN_VERIFY];  

/* The option to proceed if receiver verification fails */  
char EdiVerifyProc[EXP_MAXLEN_SN_EDI_VERIFY_PROC];  

/* The Information Exchange charging information */  
char Charge[EXP_MAXLEN_SN_CHARGE];  

/* The type of acknowledgment */  
char Ack[EXP_MAXLEN_SN_ACK];  
}
/* The retention period of the file */
char Retention[EXP_MAXLEN_SN_RETENTION];

/* The delivery service type for the file */
char Priority[EXP_MAXLEN_SN_PRIORITY];

/* Reserved field */
char Reserved1[20];

/* The type of data to send to Information Exchange */
char Datatype[EXP_MAXLEN_SN_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_SN_FORMAT];

/* The description of the file */
char Description[EXP_MAXLEN_SN_DESCRIPTION];

/* The status of data delimitation */
char Delimited[EXP_MAXLEN_SN_DELIMITED];

/* The record format of the file to be sent */
char Recformat[EXP_MAXLEN_SN_RECFORMAT];

/* The record length of the file */
char Recordlen[EXP_MAXLEN_SN_RECORDLEN];

/* The option to receive by msgclass, sender, or msgkey */
char Selectrcv[EXP_MAXLEN_SN_SELECTRCV];

/* The destination file name */
char Destfile[EXP_MAXLEN_SN_DESTFILE];

/* The destination file location */
char Destloc[EXP_MAXLEN_DESTLOC];

/* This indicates test or normal files */
char Mode[EXP_MAXLEN_SN_MODE];

/* The message name */
char MsgName[EXP_MAXLEN_SN_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_SN_MSG_SEQ_NO];

/* The translation table */
char XlateTbl[EXP_MAXLEN_SN_XLATE_TBL];

/* The option to send as EDI or unformatted */
char Sendedi[EXP_MAXLEN_SENDEDI];
Output field definitions for ExpSendOrdrReplacedRspStruct

The output can include any or all of these fields.

**OrdrName**
The order name for a send request.

| Max length: | EXP_MAXLEN_SN_ORDR_NAME (20) |
| Min length: | EXP_MINLEN_SN_ORDR_NAME (1) |
| Contents:   | Printable data. |

**OrderDescr**
The description of the send order.

| Max length: | EXP_MAXLEN_SN_ORDER_DESCR (79) |
| Min length: | EXP_MINLEN_SN_ORDER_DESCR (1) |
| Contents:   | Printable data. |

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

| Max length: | EXP_MAXLEN_SN_IE_SYSID (3) |
| Min length: | EXP_MINLEN_SN_IE_SYSID (1) |
| Contents:   | Printable data. |

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right, if needed. The first eight characters are the account ID; the next eight are the user ID.

| Max length: | EXP_MAXLEN_SN_IE_MBOX_ID (16) |
| Min length: | EXP_MINLEN_SN_IE_MBOX_ID (9) |
| Contents:   | Printable data. |
Output field definitions for ExpSendOrdrReplacedRspStruct

### TableType
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

**Max length:** EXP_MAXLEN_SN_TABLE_TYPE (1)
**Min length:** EXP_MINLEN_SN_TABLE_TYPE (1)
**Contents:** One of these valid values:
- G: Alias table for global account.
- O: Alias table for a specific account.
- P: Alias table for private use.

### AliasTbl
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

**Max length:** EXP_MAXLEN_SN_ALIAS_TBL (3)
**Min length:** EXP_MINLEN_SN_ALIAS_TBL (1)
**Contents:** Printable data.

### Aliasname
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

**Max length:** EXP_MAXLEN_SN_ALIASNAME (16)
**Min length:** EXP_MINLEN_SN_ALIASNAME (1)
**Contents:** Printable data.

### Listname
The name of the list.

**Max length:** EXP_MAXLEN_SN_LISTNAME (8)
**Min length:** EXP_MINLEN_SN_LISTNAME (1)
**Contents:** Printable data.

### Nickname
The nickname from the address book. The Nickname is specified with AddrType to determine the network address for sending and receiving.

**Max length:** EXP_MAXLEN_SN_NICKNAME (20)
**Min length:** EXP_MINLEN_SN_NICKNAME (1)
**Contents:** Printable data.
### AddrType
The address type used to send or receive data. The AddrType is specified with Nickname to determine the network address.

- **Max length:** EXP_MAXLEN_SN_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_SN_ADDR_TYPE (1)
- **Contents:** One of these valid values:
  - A Use the alias name and the alias table.
  - D Use a distribution list with the name specified in the Nickname field.
  - E Use the Information Exchange address.
  - X Use the Mail Exchange address.

### MsgClass
The user-specified Information Exchange user message class of the files to be sent. The receiver can use this name to receive only files of this particular class.

- **Max length:** EXP_MAXLEN_SN_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_SN_MSG_CLASS (1)
- **Contents:** Printable data.

### Fileid
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_SN_FILEID (54)
- **Min length:** EXP_MINLEN_SN_FILEID (1)
- **Contents:** A file system path.

### Verify
This indicates whether the verification of the receiver takes place before sending the file.

- **Max length:** EXP_MAXLEN_SN_VERIFY (1)
- **Min length:** EXP_MINLEN_SN_VERIFY (1)
- **Contents:** One of these valid values:
  - F Verify that the receiver exists. If Information Exchange cannot verify, send the data anyway.
  - N Do not verify the address.
  - Y Verify the address. If the address is invalid or cannot be verified, do not send the data.
### EdiVerifyProc
This option indicates what action to take if the address verification fails while sending envelopes from a file with multiple envelopes.

| Max length: | EXP_MAXLEN_SNEDI_VERIFYPROC (1) |
| Min length: | EXP_MINLEN_SNEDI_VERIFYPROC (1) |
| Contents:   | One of these valid values: |
|            | C  Continue processing. |
|            | S  Stop processing this file and continue with next command. |

### Charge
This indicates who pays the Information Exchange messages charges.

| Max length: | EXP_MAXLEN_SN_CHARGE (1) |
| Min length: | EXP_MINLEN_SN_CHARGE (1) |
| Contents:   | One of these valid values: |
|            | 1  The receiver pays all charges. |
|            | 2  The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges. |
|            | 3  The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default. |
|            | 4  The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges. |
|            | 5  The sender and receiver split all charges. |
|            | 6  The sender pays all charges. |
Ack
The type of acknowledgment the user wants to receive. Information Exchange places acknowledgments in the user's mailbox.

Max length: EXP_MAXLEN_SN_ACK (1)
Min length: EXP_MINLEN_SN_ACK (1)
Contents: One of these valid values:
A Information Exchange provides purge acknowledgments.
B Information Exchange provides receipt and delivery acknowledgments.
C Information Exchange provides receipt and purge acknowledgments.
D Information Exchange provides delivery acknowledgments.
E Information Exchange provides purge or delivery acknowledgments.
F Information Exchange provides receipt and either purge or delivery acknowledgments.
N Information Exchange does not provide acknowledgments.
R Information Exchange provides receipt acknowledgments.

Retention
The number of days Information Exchange keeps the file in the mailbox if no one receives it. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

Max length: EXP_MAXLEN_SN_RETENTION (3)
Min length: EXP_MINLEN_SN_RETENTION (1)
Contents: Numeric data.
Output field definitions for ExpSendOrdrReplacedRspStruct

**Priority**
The class of delivery service for the file to be sent.

Max length: EXP_MAXLEN_SN_PRIORITY (1)
Min length: EXP_MINLEN_SN_PRIORITY (1)
Contents: One of these valid values:
  - **N** Normal priority. The file is received in the order placed in the mailbox. Normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receivers’s queue.
  - **X** Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.

**Datatype**
This indicates whether the data is sent in text or binary format. Binary data is not translated to EBCDIC, and is sent as is.

Max length: EXP_MAXLEN_SN_DATATYPE (1)
Min length: EXP_MINLEN_SN_DATATYPE (1)
Contents: One of these valid values:
  - **B** Binary data.
  - **T** Text data.

**Format**
This indicates whether to send data as a file or as e-mail.

Max length: EXP_MAXLEN_SN_FORMAT (1)
Min length: EXP_MINLEN_SN_FORMAT (1)
Contents: One of these valid values:
  - **N** Do not format as e-mail.
  - **Y** Format the text data as 79-byte records, padded with blanks. The default user message class is FFMSG001.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrReplacedRspStruct

Description
This is a free-format file description.

- **Max length:** EXP_MAXLEN_SN_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_SN_DESCRIPTION (1)
- **Contents:** Printable data.

Delimited
This indicates whether or not the data is delimited with CRLF characters.

- **Max length:** EXP_MAXLEN_SN_DELIMITED_OPT (1)
- **Min length:** EXP_MINLEN_SN_DELIMITED_OPT (1)
- **Contents:** One of these valid values:
  - **N** Data is not delimited by CRLF characters.
  - **Y** Data is delimited with CRLF characters.

Recformat
The record format of the file. This information is stored in the CDH and does not affect processing on any system, except VM.

- **Max length:** EXP_MAXLEN_SN_RECFORMAT (1)
- **Min length:** EXP_MINLEN_SN_RECFORMAT (1)
- **Contents:** One of these valid values:
  - **F** Fixed-record format.
  - **N** Record format is unknown.
  - **V** Variable-record format.

Recordlen
The record length of the file to be sent. A value of 0 indicates the record length is unknown.

- **Max length:** EXP_MAXLEN_SN_RECORDLEN (5)
- **Min length:** EXP_MINLEN_SN_RECORDLEN (1)
- **Contents:** Numeric data.
**Selectrcv**
This indicates how the receiver can retrieve the file.

- **Max length:** EXP_MAXLEN_SN_SELECTRCV (1)
- **Min length:** EXP_MINLEN_SN_SELECTRCV (1)
- **Contents:** One of these valid values:
  - N The receiver can retrieve the file as usual.
  - Y The receiver must use msgkey, account ID and user ID, or class to receive this file.

**Destfile**
The original file name for the CDH. If you, as the receiver, specify the data should be received into a file with the same name as on the sender’s system, this file name is used if it is appropriate for your system. If you do not specify a destination file name, Expedite for Windows uses the file name on your system for the CDH.

- **Max length:** EXP_MAXLEN_SN_DESTFILE (54)
- **Min length:** EXP_MINLEN_SN_DESTFILE (1)
- **Contents:** A file name.

**Destloc**
The destination path for the CDH. This is provided for the receiver’s information only. If you do not specify a destination path, Expedite for Windows uses the one from your system.

- **Max length:** EXP_MAXLEN_SN_DESTLOC (65)
- **Min length:** EXP_MINLEN_SN_DESTLOC (1)
- **Contents:** A file path.

**Mode**
This indicates a test or a normal file.

- **Max length:** EXP_MAXLEN_SN_MODE (1)
- **Min length:** EXP_MINLEN_SN_MODE (1)
- **Contents:** One of these valid values:
  - N Normal mode.
  - T Test mode.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendOrdrReplacedRspStruct

**MsgName**
The user-specified message name.

- **Max length:** EXP_MAXLEN_SN_MSG_NAME (8)
- **Min length:** EXP_MINLEN_SN_MSG_NAME (1)
- **Contents:** Printable data.

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** EXP_MAXLEN_SN_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_SN_MSG_SEQ_NO (1)
- **Contents:** Printable data.

**XlateTbl**
The name of the translation table.

- **Max length:** EXP_MAXLEN_SN_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SN_XLATE_TBL (1)
- **Contents:** Printable data.

**Sendedi**
The option to send data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_SN_SENDEDI (1)
- **Min length:** EXP_MINLEN_SN_SENDEDI (1)
- **Contents:** One of these valid values:
  - **N** Send as unformatted data.
  - **Y** Send as EDI data.
ExpUsgSendOrdrFromDropBox

This function unassigns a send order from the dropoff box. If the send order name or dropoff box name specified does not exist, the return code is set to 8. If the send order is not assigned to the dropoff box, the return code is set to 4.

Function prototype ExpUsgSendOrdrFromDropBox

```c
int ExpUsgSendOrdrFromDropBox(
    struct ExpUsgSendOrdrFromDropBoxReqStruct * Input
);
```

Input arguments for ExpUsgSendOrdrFromDropBox

```c
struct ExpUsgSendOrdrFromDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The send order name */
    char OrdrName[EXP_MAXLEN_SN_ORDR_NAME];
};
```
Input field definitions for ExpUsgSendOrdrFromDropBoxReqStruct

There is no output associated with this function.

ProjName
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

DropBoxName
The dropoff box name.

- Max length: EXP_MAXLEN_DR_DROP_BOX_NAME (20)
- Min length: EXP_MINLEN_DR_DROP_BOX_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the dropoff box must exist in the database.

OrdrName
The order name for a send request.

- Max length: EXP_MAXLEN_SN_ORDR_NAME (20)
- Min length: EXP_MINLEN_SN_ORDR_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the send order must exist in the database.
Session receipts
These are the functions, the input and output structures, and the field descriptions you use to work with session receipts in the MailBox database.

ExpDeleteSessRcpt
This function deletes the session receipt and all associated order receipts from the database. If the session receipt name specified does not exist, the return code is set to 4.

Function prototype for ExpDeleteSessRcpt

```c
int ExpDeleteSessRcpt(
    struct ExpDeleteSessRcptReqStruct * Input
);
```

Input arguments for ExpDeleteSessRcpt

```c
define struct ExpDeleteSessRcptReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_SR_SRCPT_NAME];
};
```
Input field definitions for ExpDeleteSessRcptReqStruct

There is no output associated with this function.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**SrcptName**
The session receipt name.

- Max length: EXP_MAXLEN_SR_SRCPT_NAME (20)
- Min length: EXP_MINLEN_SR_SRCPT_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the session receipt must exist in the database.
ExpListSessRcpt

This function lists the session receipts on the receipt shelf in the MailBox database that correspond to the query formed in the request structure. If the query results in no session receipts found, the return code is set to 4.

Function prototype for ExpListSessRcpt

    int ExpListSessRcpt(
        struct ExpListReqStruct *Input,
        unsigned short *EntriesReturned,
        struct ExpSessRcptListedRspStruct Output
    );

Input arguments for ExpListSessRcpt

    struct ExpListReqStruct
    {
        /* The project name */
        char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

        /* The field on which to search */
        char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

        /* The search operator to use */
        char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

        /* The search value to use */
        char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

        /* The field on which to sort */
        char SortParm[EXP_MAXLEN_CF_SORT_PARM];

        /* The order in which to sort the list */
        char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

        /* The number of entries requested for receiving */
        char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

        /* The entry number at which to start the search */
        char StartIndex[EXP_MAXLEN_CF_START_INDEX];
    };

Input field definitions for ExpListSessRcptReqStruct

The input can include any or all of these fields. For a list of the output field descriptions, see “Output field definitions for ExpSessRcptListedRspStruct” on page 435.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**SearchParm**
The field on which to search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_PARM (15)
- **Min length:** EXP_MINLEN_SEARCH_PARM (1)
- **Required:** Yes, if a SearchValue is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the search fields are:
  - Name
  - Status
  - DropBoxName
  - StartDate

**SearchOp**
The operator to use for the search.

- **Max length:** EXP_MAXLEN_CF_SEARCH_OPER (2)
- **Min length:** EXP_MINLEN_SEARCH_OPER (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** Yes, the valid characters are: =, <, >, <=, >=, or <=.
**SearchValue**
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** No

**SortParm**
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the sort fields are:
  - Name
  - Status
  - DropBoxName
  - StartDate

**SortOrder**
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **RM_SORT_ASCENDING (A)**
    Sort the list in ascending order.
  - **RM_SORT_DESCENDING (D)**
    Sort the list in descending order.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListSessReptReqStruct

NumEntries
The number of entries to return in the response structure.

- Max length: EXP_MAXLEN_CF_NUM_ENTRIES (5)
- Min length: EXP_MINLEN_NUM_ENTRIES (1)
- Max value: EXP_MAXVAL_NUM_ENTRIES (65535)
- Min value: EXP_MINVAL_NUM_ENTRIES (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, for numeric data.

StartIndex
The entry number at which to start the search.

- Max length: EXP_MAXLEN_CF_START_INDEX (5)
- Min length: EXP_MINLEN_START_INDEX (1)
- Max value: EXP_MAXVAL_START_INDEX (65535)
- Min value: EXP_MINVAL_START_INDEX (1)
- Required: No
- Case sensitive: No
- Validation: Yes, for numeric data.
Output arguments for ExpListSessRcpt

```c
struct ExpSessRcptListedRspStruct
{
    /* The status code for the session receipt */
    char StatusCode[EXP_MAXLEN_SR_STATUS_CODE];

    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_SR_SRCPT_NAME];

    /* The dropoff box name for the session receipt */
    char DropBoxName[EXP_MAXLEN_DR_DROP_BOX_NAME];

    /* The starting date of the Information Exchange session */
    char StartDate[EXP_MAXLEN_SR_START_DATE];

    /* The Information Exchange session end date */
    char EndDate[EXP_MAXLEN_SR_END_DATE];

    /* The Information Exchange session key (identifier) */
    char Sesskey[EXP_MAXLEN_SR_SESSKEY];

    /* The Information Exchange version being used. */
    char Ieversion[EXP_MAXLEN_SR_IEVERSION];

    /* The Information Exchange release being used */
    char Ierelease[EXP_MAXLEN_SR_IERELEASE];

    /* The status of the last Information Exchange session */
    char LastSess[EXP_MAXLEN_SR_LAST_SESS];

    /* The response code received from Information Exchange */
    char Ierespcode[EXP_MAXLEN_SR_IERESPCODE];

    /* The start-session status */
    char StartStatus[EXP_MAXLEN_SR_START_STATUS];

    /* The end-session status */
    char EndStatus[EXP_MAXLEN_SR_END_STATUS];

    /* The session start order status */
    char StartOrdrStatus[EXP_MAXLEN_SR_START_ORDR_STATUS];

    /* The session end order status */
    char EndOrdrStatus[EXP_MAXLEN_SR_END_ORDR_STATUS];
};
```
Output field definitions for ExpSessRcptListedRspStruct

The output can include any or all of these fields.

**EntriesReturned**
The number of session receipts copied into the output structure.

**StatusCode**
The highest severity error that occurred when processing the session.

Max length: EXP_MAXLEN_SR_STATUS_CODE (5)
Min length: EXP_MINLEN_SR_STATUS_CODE (1)
Contents: Possible values include:
0  No errors occurred.
4  A warning occurred processing one or more orders.
8  One or more orders had severe errors and could not be processed.
12 A fatal error occurred and the session was incomplete.
13  The session is either in progress or was interrupted.

**SrcptName**
The session receipt name.

Max length: EXP_MAXLEN_SR_SRCPT_NAME (20)
Min length: EXP_MINLEN_SR_SRCPT_NAME (1)
Contents: Printable data.

**DropBoxName**
The dropoff box name.

Max length: EXP_MAXLEN_DR_DROP_BOX_NAME (20)
Min length: EXP_MINLEN_DR_DROP_BOX_NAME (1)
Contents: Printable data.

**StartDate**
The start date and time of the Information Exchange session.

Max length: EXP_MAXLEN_SR_START_DATE (14)
Min length: EXP_MINLEN_SR_START_DATE (14)
Contents: The date and time in YYYYMMDDHHMMSS format.
**EndDate**
The end date and time of the Information Exchange session.

- **Max length:** EXP_MAXLEN_SR_END_DATE (14)
- **Min length:** EXP_MINLEN_SR_END_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**Sesskey**
This is the Information Exchange session key.

- **Max length:** EXP_MAXLEN_SR_SESSKEY (8)
- **Min length:** EXP_MINLEN_SR_SESSKEY (1)
- **Contents:** Alphanumeric data.

**Ieverison**
The Information Exchange version that is being used.

- **Max length:** EXP_MAXLEN_SR_IEVERSION (2)
- **Min length:** EXP_MINLEN_SR_IEVERSION (1)
- **Contents:** Numeric data.

**Ierelease**
The Information Exchange release used.

- **Max length:** EXP_MAXLEN_SR_IERELEASE (2)
- **Min length:** EXP_MINLEN_SR_IERELEASE (1)
- **Contents:** Numeric data.

**LastSess**
The status of the last Information Exchange session.

- **Max length:** EXP_MAXLEN_SR_LAST_SESS (5)
- **Min length:** EXP_MINLEN_SR_LAST_SESS (1)
- **Contents:** One of these valid values:
  - 0  The last session status is successful.
  - 1  The last session status is not successful.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSessRcptListedRspStruct

**Ierespcode**
The response code received from Information Exchange.

- **Max length:** EXP_MAXLEN_SR_IERESPCODE (5)
- **Min length:** EXP_MINLEN_SR_IERESPCODE (1)
- **Contents:** Printable data.

**StartStatus**
The return code from the Information Exchange session start command.

- **Max length:** EXP_MAXLEN_SR_START_STATUS (5)
- **Min length:** EXP_MINLEN_SR_START_STATUS (1)
- **Contents:** Numeric data.

**EndStatus**
The return code from the Information Exchange session end command.

- **Max length:** EXP_MAXLEN_SR_END_STATUS (5)
- **Min length:** EXP_MINLEN_SR_END_STATUS (1)
- **Contents:** Numeric data.

**StartOrdrStatus**
The status for the session start.

- **Max length:** EXP_MAXLEN_SR_START_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_SR_START_ORDR_STATUS (1)
- **Contents:** One of these valid values:
  - P Pending
  - I Issued
  - E Complete
  - N Canceled
  - T Interrupted
  - Z Incomplete
**EndOrdrStatus**

The status for the session end.

- Max length: EXP_MAXLEN_SR_END_ORDR_STATUS (1)
- Min length: EXP_MINLEN_SR_END_ORDR_STATUS (1)
- Contents: One of these valid values:
  - P  Pending
  - I  Issued
  - E  Complete
  - N  Canceled
  - T  Interrupted
  - Z  Incomplete
Chapter 4. Writing the interface for the MailBox database

**Function prototype for ExpListRecvRcpt**

```c
int ExpListRecvRcpt(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpRecvRcptListedRspStruct *Output
);
```

**Input arguments for ExpListRecvRcpt**

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListRecvRcptReqStruct

The input can include any or all of these fields. For a list of the output field descriptions, see “Output field definitions for ExpRecvRcptListedRspStruct” on page 447.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**SearchParm**
The fields on which to search.

- Max length: EXP_MAXLEN_CF_SEARCH_PARM (15)
- Min length: EXP_MINLEN_SEARCH_PARM (1)
- Required: Yes, if a SearchValue is specified.
- Case sensitive: Yes
- Validation: Yes, the search fields are:
  - Name
  - StartDate
  - MailboxId
  - OrdrName
  - FileId
  - Class

**SearchOp**
The operator to use for the search.

- Max length: EXP_MAXLEN_CF_SEARCH_OPER (2)
- Min length: EXP_MINLEN_SEARCH_OPER (1)
- Required: Yes, if a SearchParm is specified.
- Case sensitive: No
- Validation: Yes, the valid characters are: =, <, >, <=, >=, or =<.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListRecvRcptReqStruct

**SearchValue**
The search value to use.

- **Max length:** EXP_MAXLEN_CF_SEARCH_VALUE (25)
- **Min length:** EXP_MINLEN_SEARCH_VALUE (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** No

**SortParm**
The field on which to sort.

- **Max length:** EXP_MAXLEN_CF_SORT_PARM (15)
- **Min length:** EXP_MINLEN_SORT_PARM (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the sort fields are:
  - Name
  - StartDate
  - MailboxId
  - OrdrName
  - FileId
  - Class

**SortOrder**
The order in which to sort the list.

- **Max length:** EXP_MAXLEN_CF_SORT_ORDER (1)
- **Min length:** EXP_MINLEN_SORT_ORDER (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** Yes, the valid values are:
  - RM_SORT_ASCENDING (A)
    Sort the list in ascending order.
  - RM_SORT_DESCENDING (D)
    Sort the list in descending order.
**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Chapter 4. Writing the interface for the MailBox database

Output arguments for ExpListRecvRcpt

```
struct ExpRecvRcptListedRspStruct
{
    /* The type for a receive receipt order */
    char Type[EXP_MAXLEN_DR_RCPT_TYPE];

    /* The receive receipt order */
    char RcptName[EXP_MAXLEN_XR_RCPT_NAME];

    /* The name of the file where data is stored */
    char Fileid[EXP_MAXLEN_XR_FILEID];

    /* The random ID assigned to the data */
    char Uniqueid[EXP_MAXLEN_XR_UNIQUEID];

    /* The date the received data went to Information Exchange */
    char MsgDate[EXP_MAXLEN_XR_MSG_DATE];

    /* The length of the file received */
    char Length[EXP_MAXLEN_XR_LENGTH];

    /* The Information Exchange charge information */
    char Charge[EXP_MAXLEN_XR_CHARGE];

    /* The type of acknowledgement */
    char Ack[EXP_MAXLEN_XR_ACK];

    /* The class of delivery service for this file */
    char Priority[EXP_MAXLEN_XR_PRIORITY];

    /* The option to receive data as text, binary, or EDI */
    char Datatype[EXP_MAXLEN_XR_DATATYPE];

    /* The receiver ID specified in the EDI data */
    char Receiver[EXP_MAXLEN_XR_RECEIVER];

    /* The EDI qualifier for the receiver */
    char Recvqual[EXP_MAXLEN_XR_RECVQUAL];

    /* The sender ID specified in the EDI data */
    char Sender[EXP_MAXLEN_XR_SENDER];

    /* The EDI qualifier for the sender */
    char Sendqual[EXP_MAXLEN_XR_SENDQUAL];

    /* The Interchange Control Number */
    char Controlnum[EXP_MAXLEN_XR_CONTROLNUM];
```

443
/* The type of EDI data received */
char Editype[EXP_MAXLEN_XR_EDITYPE];

/*/ The name of the system that sent the data */
char Sysname[EXP_MAXLEN_XR_SYSNAME];

/*/ The level of the system that sent the data */
char Syslevel[EXP_MAXLEN_XR_SYSLEVEL];

/*/ The original file name on the sender’s system */
char Senderfile[EXP_MAXLEN_XR_SENDERFILE];

/*/ The directory where the file was stored */
char Senderloc[EXP_MAXLEN_XR_SENDERLOC];

/*/ The description from sender */
char Description[EXP_MAXLEN_XR_DESCRIPTION];

/*/ The date and time the file was created or last modified */
char Filedate[EXP_MAXLEN_XR_FILEDATE];

/*/ The message sequence number */
char IeMsgSeqNo[EXP_MAXLEN_XR_IE_MSG_SEQNO];

/*/ The session access key used */
char Sessionkey[EXP_MAXLEN_XR_SESSIONKEY];

/*/ The archive ID used */
char Archiveid[EXP_MAXLEN_XR_ARCHIVEID];

/*/ The record format of the file on the sender’s system */
char Recfm[EXP_MAXLEN_XR_RECFM];

/*/ The original record length of the file */
char Reclen[EXP_MAXLEN_XR_RECLEN];

/*/ The method used to delimit the records */
char Recdlm[EXP_MAXLEN_XR_RECDLM];

/*/ The way record delimiters were processed */
char Delimited[EXP_MAXLEN_XR_DELIMITED];

/*/ The type of system that sent the data */
char Systype[EXP_MAXLEN_XR_SYSTYPE];

/*/ The software version of the system sending the data */
char Sysver[EXP_MAXLEN_XR_SYSVER];
/* The name of the software compression application */
char Comsw[EXP_MAXLEN_XR_COMSW];

/* The version of the software compression */
char Comver[EXP_MAXLEN_XR_COMVER];

/* The name of the compressed file */
char Comfile[EXP_MAXLEN_XR_COMFILE];

/* The code page used by the sending system */
char Codepage[EXP_MAXLEN_XR_CODEPAGE];

/* The decompression return code */
char Dcmprc[EXP_MAXLEN_XR_DCMPRC];

/* The indicator for test or normal files */
char Mode[EXP_MAXLEN_XR_MODE];

/* The Information Exchange alias-table type */
char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

/* The Information Exchange alias table */
char AliasTbl[EXP_MAXLEN_XR_ALIAS_TBL];

/* The Information Exchange alias name */
char Aliasname[EXP_MAXLEN_XR_ALIASNAME];

/* The Information Exchange system ID */
char IeSysid[EXP_MAXLEN_XR_IE_SYSID];

/* The Information Exchange account ID and user ID */
char IeMboxId[EXP_MAXLEN_XR_IE_MBOX_ID];

/* The Information Exchange message class */
char MsgClass[EXP_MAXLEN_XR_MSG_CLASS];

/* The message name */
char MsgName[EXP_MAXLEN_XR_MSG_NAME];

/* The unique number assigned to the data */
char MsgSeqNo[EXP_MAXLEN_XR_MSG_SEQ_NO];

/* The translation table */
char XlateTbl[EXP_MAXLEN_XR_XLATE_TBL];

/* The status of how an order was processed */
char StatusCode[EXP_MAXLEN_XR_STATUS_CODE];

/* The Expedite for Windows session start date */
char PstartDate[EXP_MAXLEN_XR_PSTART_DATE];
/*The Expedite for Windows session end date */
char PendDate[EXP_MAXLEN_XR_PEND_DATE];

/* The order processed */
char OrderName[EXP_MAXLEN_XR_ORDER_NAME];

/* The commit code used */
char CmmtCode[EXP_MAXLEN_XR_CMMT_CODE];

/* The sequence number for this receipt */
char SequenceNum[EXP_MAXLEN_XR_SEQUENCE_NUM];

/* The order status */
char OrdrStatus[EXP_MAXLEN_XR_ORDR_STATUS];

/* The listname used for Information Exchange temporary list */
char Listname[EXP_MAXLEN_XR_LISTNAME];

/* The nickname specified on order */
char Nickname[EXP_MAXLEN_XR_NICKNAME];

/* The address type specified on order */
char AddrType[EXP_MAXLEN_XR_ADDR_TYPE];
};
Output field definitions for ExpRecvRcptListedRspStruct

The output can include any or all of these fields.

EntriesReturned
The number of receive receipts copied into the output structure.

Type
The receive receipt type.

Max length: EXP_MAXLEN_XR_TYPE (1)
Min length: EXP_MINLEN_XR_TYPE (1)
Contents: Contains the character R.

RcptName
The receive receipt name.

Max length: EXP_MAXLEN_XR_RCPT_NAME (20)
Min length: EXP_MINLEN_XR_RCPT_NAME (1)
Contents: Printable data.

Fileid
The name of the file where the received data was stored.

Max length: EXP_MAXLEN_XR_FILEID (65)
Min length: EXP_MINLEN_XR_FILEID (1)
Contents: A file path.

Uniqueid
The ID assigned to the data by the sending interface. Use it to identify and associate the data with any acknowledgments received.

Max length: EXP_MAXLEN_XR_UNIQUEID (8)
Min length: EXP_MINLEN_XR_UNIQUEID (1)
Contents: Printable data.
**MsgDate**
The date the data was received and placed into Information Exchange.

- **Max length:** EXP_MAXLEN_XR_MSG_DATE (14)
- **Min length:** EXP_MINLEN_XR_MSG_DATE (14)
- **Contents:** Printable data.

**Length**
The length of the file received.

- **Max length:** EXP_MAXLEN_XR_LENGTH (9)
- **Min length:** EXP_MINLEN_XR_LENGTH (1)
- **Contents:** Numeric data.

**Charge**
This indicates who pays the Information Exchange message charges.

- **Max length:** EXP_MAXLEN_XR_CHARGE (1)
- **Min length:** EXP_MINLEN_XR_CHARGE (1)
- **Contents:** One of these valid values:
  1. The receiver pays all charges.
  2. The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  3. The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  4. The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  5. The sender and receiver split all charges.
  6. The sender pays all charges.
**Ack**
The type of acknowledgment the user wants to receive. Information Exchange places acknowledgments into the user’s mailbox.

<table>
<thead>
<tr>
<th>Max length:</th>
<th>EXP_MAXLEN_XR_ACK (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min length:</td>
<td>EXP_MINLEN_XR_ACK (1)</td>
</tr>
<tr>
<td>Contents:</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td>A</td>
<td>Information Exchange provides purge acknowledgments.</td>
</tr>
<tr>
<td>B</td>
<td>Information Exchange provides receipt and delivery acknowledgments.</td>
</tr>
<tr>
<td>C</td>
<td>Information Exchange provides receipt and purge acknowledgments.</td>
</tr>
<tr>
<td>D</td>
<td>Information Exchange provides delivery acknowledgments.</td>
</tr>
<tr>
<td>E</td>
<td>Information Exchange provides purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>F</td>
<td>Information Exchange provides receipt and either purge or delivery acknowledgments.</td>
</tr>
<tr>
<td>N</td>
<td>Information Exchange does not provide acknowledgments.</td>
</tr>
<tr>
<td>R</td>
<td>Information Exchange provides receipt acknowledgments.</td>
</tr>
</tbody>
</table>
**Priority**
The class of delivery service for this file.

- **Max length:** EXP_MAXLEN_XR_PRIORITY (1)
- **Min length:** EXP_MINLEN_XR_PRIORITY (1)
- **Contents:** One of these valid values:
  - **N** Normal priority. The file is received in the order placed in the mailbox. Normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express priority. Express delivery to those users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Datatype**
This indicates whether the data is text, binary, or EDI.

- **Max length:** EXP_MAXLEN_XR_DATATYPE (1)
- **Min length:** EXP_MINLEN_XR_DATATYPE (1)
- **Contents:** One of these valid values:
  - **B** Binary data.
  - **E** Supported EDI formatted data.
  - **T** Text data.

**Receiver**
The receiver ID specified in the EDI data.

- **Max length:** EXP_MAXLEN_XR_RECEIVER (35)
- **Min length:** EXP_MINLEN_XR_RECEIVER (1)
- **Contents:** Printable data.

**Recvqual**
The EDI qualifier for the receiver specified in the EDI data.

- **Max length:** EXP_MAXLEN_XR_RECVQUAL (4)
- **Min length:** EXP_MINLEN_XR_RECVQUAL (1)
- **Contents:** Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvReptListedRspStruct

Sender
The sender ID specified in the EDI data.

Max length: EXP_MAXLEN_XR_SENDER (35)
Min length: EXP_MINLEN_XR_SENDER (1)
Contents: Printable data.

Sendqual
The EDI qualifier for the sender specified in the EDI data.

Max length: EXP_MAXLEN_XR_SENDQUAL (4)
Min length: EXP_MINLEN_XR_SENDQUAL (1)
Contents: Printable data.

Controlnum
The Interchange Control Number from the X12 or UCS data. For UN/TDI data, this is the SNRF element. For EDIFACT data, this is the element 0020 (Interchange Control Reference).

Max length: EXP_MAXLEN_XR_CONTROLNUM (14)
Min length: EXP_MINLEN_XR_CONTROLNUM (1)
Contents: Printable data.

Editype
The type of EDI data received: X12, UCS, UN/TDI, EDIFACT, or unformatted.

Max length: EXP_MAXLEN_XR_EDITYPE (11)
Min length: EXP_MINLEN_XR_EDITYPE (3)
Contents: Printable data.

Sysname
This indicates the name of the system that sent the data.

Max length: EXP_MAXLEN_XR_SYSNAME (8)
Min length: EXP_MINLEN_XR_SYSNAME (1)
Contents: Printable data.
**Syslevel**
The system level of the system that sent the data.

- **Max length**: EXP_MAXLEN_XR_SYSLEVEL (8)
- **Min length**: EXP_MINLEN_XR_SYSLEVEL (1)
- **Contents**: Printable data.

**Senderfile**
The original file name on the sender's system.

- **Max length**: EXP_MAXLEN_XR_SENDERFILE (54)
- **Min length**: EXP_MINLEN_XR_SENDERFILE (1)
- **Contents**: Printable data.

**Senderloc**
If the sending system was a workstation, this contains the directory where the file was stored on the sender's system.

- **Max length**: EXP_MAXLEN_XR_SENDERLOC (65)
- **Min length**: EXP_MINLEN_XR_SENDERLOC (1)
- **Contents**: A file path.

**Description**
The free-format data description written by the sender.

- **Max length**: EXP_MAXLEN_XR_DESCRIPTION (79)
- **Min length**: EXP_MINLEN_XR_DESCRIPTION (1)
- **Contents**: Printable data.

**Filedate**
The date the file was created or last edited on the sender's system.

- **Max length**: EXP_MAXLEN_XR_FILEDATE (14)
- **Min length**: EXP_MINLEN_XR_FILEDATE (14)
- **Contents**: The date and time in YYYYMMDDHHMMSS format.

**IeMsgSeqNo**
The message sequence number assigned by Information Exchange.

- **Max length**: EXP_MAXLEN_XR_IE_MSG_SEQ_NO (5)
- **Min length**: EXP_MINLEN_XR_IE_MSG_SEQ_NO (1)
- **Contents**: Printable data.
### Sessionkey
The session access key used when the file was received.

- **Max length:** EXP_MAXLEN_XR_SESSIONKEY (8)
- **Min length:** EXP_MINLEN_XR_SESSIONKEY (8)
- **Contents:** Printable data.

### Recfm
The record format of the file on the sender's system.

- **Max length:** EXP_MAXLEN_XR_RECFM (4)
- **Min length:** EXP_MINLEN_XR_RECFM (1)
- **Contents:** Printable data.

### Reclen
The file's original record length.

- **Max length:** EXP_MAXLEN_XR_RECLEN (5)
- **Min length:** EXP_MINLEN_SN_RECLEN (1)
- **Max value:** EXP_MAXVAL_SN_RECORDLEN (65535)
- **Min value:** EXP_MINVAL_SN_RECORDLEN (1)
- **Contents:** Numeric data.

### Recdlm
The way the record delimiters were processed when the file was received.

- **Max length:** EXP_MAXLEN_XR_RECDLM (1)
- **Min length:** EXP_MINLEN_XR_RECDLM (1)
- **Contents:** One of these valid values:
  - C Convert the line-length delimiters to CRLF delimiters.
  - I Ignore line-length delimiters.
  - R Remove line-length delimiters.
**Delimited**
Indicates whether the data contains carriage-return line-feed delimiters.

- **Max length:** EXP_MAXLEN_XR_DELIMITED (1)
- **Min length:** EXP_MINLEN_XR_DELIMITED (1)
- **Contents:** One of these valid values:
  - N Data is not delimited by CRLF characters.
  - Y Data is delimited with CRLF characters.

**Systype**
The type of system that sent the data.

- **Max length:** EXP_MAXLEN_XR_SYSTYPE (2)
- **Min length:** EXP_MINLEN_XR_SYSTYPE (2)
- **Contents:** Printable data.

**Sysver**
The software version of the Expedite for Windows system sending the data.

- **Max length:** EXP_MAXLEN_XR_SYSVER (3)
- **Min length:** EXP_MINLEN_XR_SYSVER (1)
- **Contents:** Printable data.

**Comsw**
The compression software package name used to compress the file.

- **Max length:** EXP_MAXLEN_XR_COMSW (10)
- **Min length:** EXP_MINLEN_XR_COMSW (1)
- **Contents:** Printable data.

**Comver**
The version of the software compression package used to compress the file.

- **Max length:** EXP_MAXLEN_XR_COMVER (5)
- **Min length:** EXP_MINLEN_XR_COMVER (1)
- **Contents:** Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvRecptListedRspStruct

Comfile
The name of the compressed file.

- Max length: \texttt{EXP\_MAXLEN\_XR\_COMFILE} (54)
- Min length: \texttt{EXP\_MINLEN\_XR\_COMFILE} (1)
- Contents: A file name.

Codepage
The code page used by the sending system to determine the character representation of a given byte.

- Max length: \texttt{EXP\_MAXLEN\_XR\_CODEPAGE} (3)
- Min length: \texttt{EXP\_MINLEN\_XR\_CODEPAGE} (3)
- Contents: Printable data.

Dcmprc
The decompression return code.

- Max length: \texttt{EXP\_MAXLEN\_XR\_DCMPRC} (5)
- Min length: \texttt{EXP\_MINLEN\_XR\_DCMPRC} (1)
- Contents: Printable data.

Mode
This indicates a test or a normal file.

- Max length: \texttt{EXP\_MAXLEN\_XR\_MODE} (1)
- Min length: \texttt{EXP\_MINLEN\_XR\_MODE} (1)
- Contents: One of these valid values:
  - \texttt{N} Normal mode.
  - \texttt{T} Test mode.

TableType
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

- Max length: \texttt{EXP\_MAXLEN\_XR\_TABLE\_TYPE} (1)
- Min length: \texttt{EXP\_MINLEN\_XR\_TABLE\_TYPE} (1)
- Contents: One of these valid values:
  - \texttt{G} Global alias table.
  - \texttt{O} Alias table for a specific account.
  - \texttt{P} Alias table for private use.
AliasTbl
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_XR_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_XR_ALIAS_TBL (1)
- **Contents:** Printable data.

Aliasname
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_XR_ALIASNAME (16)
- **Min length:** EXP_MINLEN_XR_ALIASNAME (1)
- **Contents:** Printable data.

IeSysid
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_XR_IE_SYSID (3)
- **Min length:** EXP_MINLEN_XR_IE_SYSID (1)
- **Contents:** Printable data.

IeMboxId
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_XR_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_XR_IE_MBOX_ID (9)
- **Contents:** Printable data.

MsgClass
The Information Exchange user message class specified by the sender.

- **Max length:** EXP_MAXLEN_XR_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_XR_MSG_CLASS (1)
- **Contents:** Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpRecvRcptListedRspStruct

**MsgName**
The user-specified message name.

- **Max length:** EXP_MAXLEN_XR_MSG_NAME (8)
- **Min length:** EXP_MINLEN_XR_MSG_NAME (1)
- **Contents:** Printable data.

**MsgSeqNo**
The user-specified message sequence number.

- **Max length:** EXP_MAXLEN_XR_MSG_SEQ_NO (5)
- **Min length:** EXP_MINLEN_XR_MSG_SEQ_NO (1)
- **Contents:** Printable data.

**XlateTbl**
The translation table name.

- **Max length:** EXP_MAXLEN_XR_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_XR_XLATE_TBL (1)
- **Contents:** Printable data.

**StatusCode**
The status results of processing a receipt order.

- **Max length:** EXP_MAXLEN_XR_STATUS_CODE (5)
- **Min length:** EXP_MINLEN_XR_STATUS_CODE (1)
- **Contents:** Numeric data.

**PstartDate**
The start date and time for the Expedite for Windows session.

- **Max length:** EXP_MAXLEN_XR_PSTART_DATE (14)
- **Min length:** EXP_MINLEN_XR_PSTART_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.

**PendDate**
The end date and time for the Expedite for Windows session.

- **Max length:** EXP_MAXLEN_XR_PEND_DATE (14)
- **Min length:** EXP_MINLEN_XR_PEND_DATE (14)
- **Contents:** The date and time in YYYYMMDDHHMMSS format.
**OrdrName**
The receive order name that produced the receive receipt.

Max length: EXP_MAXLEN_XR_ORDR_NAME (20)
Min length: EXP_MINLEN_XR_ORDR_NAME (1)
Contents: Printable data.

**SequenceNum**
If multiple files are received, this is the sequence number assigned by Expedite for Windows. Sequence numbers start with 1 for each receive order and are incremented each time a new file is received for that order.

Max length: EXP_MAXLEN_XR_SEQUENCE_NUM (20)
Min length: EXP_MINLEN_XR_SEQUENCE_NUM (1)
Contents: Numeric data.

**OrdrStatus**
The processing status of the order. This field is included only on the first receipt produced for a receive order. If multiple files are received, only the first receipt shows the order status.

Max length: EXP_MAXLEN_XR_ORDR_STATUS (1)
Min length: EXP_MINLEN_XR_ORDR_STATUS (1)
Contents: One of these valid values:
- **C** The order processing is complete.
- **I** The order has been issued, but processing is not complete.
- **P** The order is pending and has not been issued.

**Listname**
The temporary list name generated to define the list to Information Exchange.

Max length: EXP_MAXLEN_XR_LISTNAME (8)
Min length: EXP_MINLEN_XR_LISTNAME (1)
Contents: Alphanumeric data.
**Nickname**

The nickname specified on the receive order.

- **Max length:** EXP_MAXLEN_XR_NICKNAME (20)
- **Min length:** EXP_MINLEN_XR_NICKNAME (1)
- **Contents:** Alphanumeric data.

**AddrType**

The address type specified on the receive order.

- **Max length:** EXP_MAXLEN_XR_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_XR_ADDR_TYPE (1)
- **Contents:** The valid values are:
  - **A** The alias name and the alias table.
  - **D** A distribution list with the name specified in the Nickname field.
  - **E** The Information Exchange address.
  - **X** The Mail Exchange address.
Send receipts
This section describes the functions, the input and output structure, and the field descriptions you use to work with send receipts in the MailBox database.

ExpListSendRcpt
This function lists the send receipts on the receipt shelf in the Mailbox database that correspond to the query formed in the request structure. If the query results in no send receipts found, the return code is set to 4.

Function prototype for ExpListSendRcpt

```c
int ExpListSendRcpt(
    struct ExpListReqStruct *Input,
    unsigned short *EntriesReturned,
    struct ExpSendRcptListedRspStruct *Output
);
```

Input arguments for ExpListSendRcpt

```c
struct ExpListReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The field on which to search */
    char SearchParm[EXP_MAXLEN_CF_SEARCH_PARM];

    /* The search operator to use */
    char SearchOp[EXP_MAXLEN_CF_SEARCH_OP];

    /* The search value to use */
    char SearchValue[EXP_MAXLEN_CF_SEARCH_VALUE];

    /* The field on which to sort */
    char SortParm[EXP_MAXLEN_CF_SORT_PARM];

    /* The order in which to sort the list */
    char SortOrder[EXP_MAXLEN_CF_SORT_ORDER];

    /* The number of entries requested for receiving */
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpListSendRcptReqStruct

For a list of the output field descriptions, see “Output field definitions for ExpSendRcptListedRspStruct” on page 466.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_PJ_PROJ_NAME (20)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_PJ_PROJ_NAME (1)</td>
</tr>
<tr>
<td>Required</td>
<td>Yes</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>Yes, the project must be open.</td>
</tr>
</tbody>
</table>

**SearchParm**
The field on which to search.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_CF_SEARCH_PARM (15)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_SEARCH_PARM (1)</td>
</tr>
<tr>
<td>Required</td>
<td>Yes, if a SearchValue is specified.</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>Yes, the search fields are:</td>
</tr>
<tr>
<td></td>
<td>• Name</td>
</tr>
<tr>
<td></td>
<td>• StartDate</td>
</tr>
<tr>
<td></td>
<td>• OrdrName</td>
</tr>
<tr>
<td></td>
<td>• MailboxId</td>
</tr>
<tr>
<td></td>
<td>• Class</td>
</tr>
</tbody>
</table>

**SearchOp**
The operator to use for the search.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_CF_SEARCH_OPER (2)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_SEARCH_OPER (1)</td>
</tr>
<tr>
<td>Required</td>
<td>Yes, if a SearchParm is specified.</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>Yes, the valid characters are: =, &lt;, &gt;, &lt;=, &gt;=, or &lt;=.</td>
</tr>
</tbody>
</table>
**SearchValue**
The search value to use.

- **Max length:** \(\text{EXP\_MAXLEN\_CF\_SEARCH\_VALUE}\) (25)
- **Min length:** \(\text{EXP\_MINLEN\_SEARCH\_VALUE}\) (1)
- **Required:** Yes, if a SearchParm is specified.
- **Case sensitive:** No
- **Validation:** No

**SortParm**
The field on which to sort.

- **Max length:** \(\text{EXP\_MAXLEN\_CF\_SORT\_Parm}\) (15)
- **Min length:** \(\text{EXP\_MINLEN\_SORT\_Parm}\) (1)
- **Required:** Yes, if a SortOrder is specified.
- **Case sensitive:** Yes
- **Validation:** Yes, the sort fields are:
  - Name
  - StartDate
  - OrdrName
  - MailboxId
  - Class

**SortOrder**
The order in which to sort the list.

- **Max length:** \(\text{EXP\_MAXLEN\_CF\_SORT\_ORDER}\) (1)
- **Min length:** \(\text{EXP\_MINLEN\_SORT\_ORDER}\) (1)
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** Yes, the valid values are:
  - **RM\_SORT\_ASCENDING** (A)
    Sort the list in ascending order.
  - **RM\_SORT\_DESCENDING** (D)
    Sort the list in descending order.
Chapter 4. Writing the interface for the MailBox database

Input field definitions for ExpListSendReptReqStruct

**NumEntries**
The number of entries to return in the response structure.

| Max length: | EXP_MAXLEN_CF_NUM_ENTRIES (5) |
| Min length: | EXP_MINLEN_NUM_ENTRIES (1) |
| Max value:  | EXP_MAXVAL_NUM_ENTRIES (65535) |
| Min value:  | EXP_MINVAL_NUM_ENTRIES (1) |
| Required:   | Yes |
| Case sensitive: | No |
| Validation: | Yes, for numeric data. |

**StartIndex**
The entry number at which to start the search.

| Max length: | EXP_MAXLEN_CF_START_INDEX (5) |
| Min length: | EXP_MINLEN_START_INDEX (1) |
| Max value:  | EXP_MAXVAL_START_INDEX (65535) |
| Min value:  | EXP_MINVAL_START_INDEX (1) |
| Required:   | No |
| Case sensitive: | No |
| Validation: | Yes, for numeric data. |
Output arguments for ExpListSendRcpt

```c
struct ExpSendRcptListedRspStruct
{
    /* The type for a send receipt */
    char RcptType[EXP_MAXLEN_DR_RCPT_TYPE];

    /* The send-receipt name */
    char RcptName[EXP_MAXLEN_XS_RCPT_NAME];

    /* The unique ID of the file sent */
    char Uniqueid[EXP_MAXLEN_XS_UNIQUEID];

    /* The length of the file or EDI envelope */
    char Length[EXP_MAXLEN_XS_LENGTH];

    /* The type of EDI data sent */
    char Editype[EXP_MAXLEN_XS_EDITYPE];

    /* The destination specified in the EDI data */
    char Destination[EXP_MAXLEN_XS_DESTINATION];

    /* The EDI qualifier for the destination */
    char Qualifier[EXP_MAXLEN_XS_QUALIFIER];

    /* The interchange control number for X12 and other formats */
    char Controlnum[EXP_MAXLEN_XS_CONTROLNUM];

    /* The Information Exchange alias-table type */
    char TableType[EXP_MAXLEN_RV_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_XS_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_XS_ALIASNAME];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_XS_IE_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_XS_IE_MBOX_ID];

    /* The list name */
    char Listname[EXP_MAXLEN_XS_LISTNAME];

    /* The Information Exchange message class */
    char MsgClass[EXP_MAXLEN_XS_MSG_CLASS];
};
```
/** The message name */
char MsgName[EXP_MAXLEN_XS_MSG_NAME];

/** The message sequence number */
char MsgSeqNo[EXP_MAXLEN_XS_MSG_SEQ_NO];

/** The status of how an order was processed */
char StatusCode[EXP_MAXLEN_XS_STATUS_CODE];

/** The Expedite for Windows session start date */
char PstartDate[EXP_MAXLEN_XS_PSTART_DATE];

/** The Expedite for Windows session end date */
char PendDate[EXP_MAXLEN_XS_PEND_DATE];

/** The name for a send order */
char OrdrName[EXP_MAXLEN_XS_ORDR_NAME];

/** The commit code used */
char CmmtCode[EXP_MAXLEN_XS_CMMT_CODE];

/** The sequence number */
char SequenceNum[EXP_MAXLEN_XS_SEQUENCE_NUM];

/** The file sent */
char FileId[EXP_MAXLEN_XS_FILEID];

/** The order processing status */
char OrdrStatus[EXP_MAXLEN_XS_ORDR_STATUS];

/** The nickname specified on the send order */
char Nickname[EXP_MAXLEN_XS_NICKNAME];

/** The address type specified on the send order */
char AddrType[EXP_MAXLEN_XS_ADDR_TYPE];

/** This indicates if the file was sent: Y or N */
char SentCode[EXP_MAXLEN_XS_SENT_CODE];
};
Output field definitions for ExpSendRcptListedRspStruct

The response can include any or all of these fields.

**EntriesReturned**
The number of send receipts copied into the output structure.

**Type**
The send receipt type.

- **Max length:** EXP_MAXLEN_XS_TYPE (1)
- **Min length:** EXP_MINLEN_XS_TYPE (1)
- **Contents:** Contains the character S.

**RcptName**
The send receipt key.

- **Max length:** EXP_MAXLEN_XS_RCPT_NAME (20)
- **Min length:** EXP_MINLEN_XS_RCPT_NAME (1)
- **Contents:** Printable data.

**Uniqueid**
The ID that the sending interface assigned to the data. Use it to identify the data and associate the data with any acknowledgments received.

- **Max length:** EXP_MAXLEN_XS_UNIQUEID (8)
- **Min length:** EXP_MINLEN_XS_UNIQUEID (1)
- **Contents:** Printable data.

**Length**
The length of the file or EDI envelope.

- **Max length:** EXP_MAXLEN_XS_LENGTH (9)
- **Min length:** EXP_MINLEN_XS_LENGTH (1)
- **Contents:** Numeric data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendRcptListedRspStruct

**Editype**
The type of EDI data sent. The type can be one of the following: X12, UCS, UN/TDI, or EDIFACT.

- Max length: EXP_MAXLEN_XS_EDITYPE (7)
- Min length: EXP_MINLEN_XS_EDITYPE (1)
- Contents: Printable data.

**Destination**
The destination specified in the EDI data.

- Max length: EXP_MAXLEN_XS_DESTINATION (35)
- Min length: EXP_MINLEN_XS_DESTINATION (1)
- Contents: Printable data.

**Qualifier**
The EDI qualifier for the destination.

- Max length: EXP_MAXLEN_XS_QUALIFIER (4)
- Min length: EXP_MINLEN_XS_QUALIFIER (1)
- Contents: Printable data.

**Controlnum**
The interchange control number from the X12 or UCS data. For UN/TDI data, this is the SNRF element. For EDIFACT data, it is the data element 0020 (Interchange Control Reference).

- Max length: EXP_MAXLEN_XS_CONTROLNUM (14)
- Min length: EXP_MINLEN_XS_CONTROLNUM (1)
- Contents: Printable data.

**TableType**
The alias-table type for an Information Exchange alias table. The TableType is specified with AliasTbl and Aliasname.

- Max length: EXP_MAXLEN_TABLE_TYPE (1)
- Min length: EXP_MINLEN_TABLE_TYPE (1)
- Contents: One of these valid values:
  - G  Global alias table.
  - O  Alias table for a specific account.
  - P  Alias table for private use.
**AliasTbl**  
The alias-table name for an Information Exchange alias table. The AliasTbl is specified with TableType and Aliasname.

- **Max length:** EXP_MAXLEN_ALIAS_TBL (3)  
- **Min length:** EXP_MINLEN_ALIAS_TBL (1)  
- **Contents:** Printable data.

**Aliasname**  
The alias name in an Information Exchange alias table. The Aliasname is specified with TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_ALIASNAME (16)  
- **Min length:** EXP_MINLEN_ALIASNAME (1)  
- **Contents:** Printable data.

**IeSysid**  
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_XS_IE_SYSID (3)  
- **Min length:** EXP_MINLEN_XS_IE_SYSID (1)  
- **Contents:** Printable data.

**IeMboxId**  
The Information Exchange account ID and user ID. Each value is specified with eight characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_XS_IE_MBOX_ID (16)  
- **Min length:** EXP_MINLEN_XS_IE_MBOX_ID (9)  
- **Contents:** Printable data.

**Listname**  
The list name where the data was sent.

- **Max length:** EXP_MAXLEN_XS_LISTNAME (8)  
- **Min length:** EXP_MINLEN_XS_LISTNAME (1)  
- **Contents:** Printable data.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendRcptListedRspStruct

**MsgClass**
The Information Exchange user message class specified by the user.

Max length: \( \text{EXP\_MAXLEN\_XS\_MSG\_CLASS} \) (8)
Min length: \( \text{EXP\_MINLEN\_XS\_MSG\_CLASS} \) (1)
Contents: Printable data.

**MsgName**
The user-specified message name.

Max length: \( \text{EXP\_MAXLEN\_XS\_MSG\_NAME} \) (8)
Min length: \( \text{EXP\_MINLEN\_XS\_MSG\_NAME} \) (1)
Contents: Printable data.

**MsgSeqNo**
The user-specified message sequence number.

Max length: \( \text{EXP\_MAXLEN\_XS\_MSG\_SEQ\_NO} \) (5)
Min length: \( \text{EXP\_MINLEN\_XS\_MSG\_SEQ\_NO} \) (1)
Contents: Printable data.

**StatusCode**
The status results of processing a receipt order.

Max length: \( \text{EXP\_MAXLEN\_XS\_STATUS\_CODE} \) (5)
Min length: \( \text{EXP\_MINLEN\_XS\_STATUS\_CODE} \) (1)
Contents: Numeric data.

**PstartDate**
The start and time for the Expedite for Windows session.

Max length: \( \text{EXP\_MAXLEN\_XS\_PSTART\_DATE} \) (14)
Min length: \( \text{EXP\_MINLEN\_XS\_PSTART\_DATE} \) (14)
Contents: The date and time in YYYYMMDDHHMMSS format.

**PEndDate**
The end date and time for the Expedite for Windows session.

Max length: \( \text{EXP\_MAXLEN\_XS\_PEND\_DATE} \) (14)
Min length: \( \text{EXP\_MINLEN\_XS\_PEND\_DATE} \) (14)
Contents: The date and time in YYYYMMDDHHMMSS format.
Output field definitions for ExpSendRcptListedRspStruct

**OrdrName**
The send order name that produced the send receipt.

- **Max length:** EXP_MAXLEN_SN_ORDR_NAME (20)
- **Min length:** EXP_MINLEN_SN_ORDR_NAME (1)
- **Contents:** Printable data.

**SequenceNum**
If multiple files are received, this field indicates the sequence number assigned by Expedite for Windows. Sequence numbers start with 1 for each receive order and are incremented each time a new file is received for that order.

- **Max length:** EXP_MAXLEN_XS_SEQUENCE_NUM (20)
- **Min length:** EXP_MINLEN_XS_SEQUENCE_NUM (1)
- **Contents:** Numeric data.

**Fileid**
The file name where the received data is stored.

- **Max length:** EXP_MAXLEN_XS_FILEID (1)
- **Min length:** EXP_MINLEN_XS_FILEID (1)
- **Contents:** A file path.

**OrdrStatus**
The processing status of the order. This field is included only on the first receipt produced for a receive order. If multiple files are received, only the first receipt shows the order status.

- **Max length:** EXP_MAXLEN_XS_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_XS_ORDR_STATUS (1)
- **Contents:** One of these valid values:
  - C The order processing is complete.
  - I The order has been issued, but processing is not complete.
  - P The order is pending and has not been issued.
Chapter 4. Writing the interface for the MailBox database

Output field definitions for ExpSendRcptListedRspStruct

**Nickname**
The nickname specified on the receive order.

- **Max length:** EXP_MAXLEN_XS_NICKNAME (20)
- **Min length:** EXP_MINLEN_XS_NICKNAME (1)
- **Contents:** Alphanumeric data.

**AddrType**
The address type specified on the receive order.

- **Max length:** EXP_MAXLEN_XS_ADDR_TYPE (1)
- **Min length:** EXP_MINLEN_XS_ADDR_TYPE (1)
- **Contents:** The valid values are:
  - A  The alias name and the alias table.
  - D  A distribution list with the name specified in the Nickname field.
  - E  The Information Exchange address.
  - X  The Mail Exchange address.
Output field definitions for ExpSendRcptListedRspStruct
Chapter 5

Writing the interface for Information Exchange

Using Expedite for Windows, you can communicate with Information Exchange as a messaging agent to perform any of the following tasks:

- Viewing messages
- Processing dropoff boxes
- Receiving orders
- Sending orders

The database sections and the related function discussed in this chapter are:

Communicating with Information Exchange ........................................ 474
  ExpOpenMSys ................................................................. 474
  ExpCloseMSys .............................................................. 475

Processing dropoff boxes in batch mode ................................. 476
  ExpProcDropBox ............................................................ 476

Starting and ending Information Exchange sessions ..................... 481
  ExpEndSess ................................................................. 481
  ExpStartSess ............................................................... 487

Recovering Information Exchange sessions ............................... 498
  ExpRecoverSessRept ...................................................... 498

Interactive session commands .................................................. 504
  ExpDoPurgOrdr ............................................................. 504
  ExpDoQuryOrdr ............................................................ 507
  ExpDoRecvOrdr ............................................................. 518
  ExpDoSendOrdr ............................................................. 536
Communicating with Information Exchange

This chapter describes the functions, the input and output arguments, and the fields used when communicating with Information Exchange.

ExpOpenMSys

This function opens an Information Exchange message system agent so that you can work with Information Exchange in batch mode or in interactive mode.

Function prototype for ExpOpenMSys

```c
int ExpOpenMSys(
    struct ExpOpenMSysReqStruct * Input
);
```

Input arguments for ExpOpenMSys

```c
struct ExpOpenMSysReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpOpenMSysReqStruct

There is no output associated with this function.

ProjName

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.
Chapter 5. Writing the interface for Information Exchange

Function prototype for ExpCloseMSys

```c
int ExpCloseMSys(
    struct ExpCloseMSysReqStruct * Input
);
```

Input arguments for ExpCloseMSys

```c
struct ExpCloseMSysReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];
};
```

Input field definitions for ExpCloseMSysReqStruct

There is no output associated with this function.

**ProjName**

The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.
Processing dropoff boxes in batch mode

**ExpProcDropBox**

This function processes a dropoff box in batch mode. The function call does not return control to your application until dropoff box processing is complete. The output structure contains the information from the session receipt. The session receipt is similar to the SESSIONEND record in Expedite Base. To see the results of each order’s processing, list the order receipts on the session receipt.

If there are no orders assigned to the dropoff box, the return code is set to 8.

**Function prototype for ExpProcDropBox**

```c
int ExpProcDropBox(
    struct ExpProcDropBoxReqStruct * Input,
    struct ExpDropBoxProcRspStruct * Output
);
```

**Input arguments for ExpProcDropBox**

```c
struct ExpProcDropBoxReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_SS_DROP_BOX_NAME];

    /* The connection status indicator */
    char ConnectStat[EXP_MAXLEN_SS_CONNECT_STAT];
};
```
Input field definitions for ExpProcDropBoxReqStruct

For a list of the output fields, see “Output field definitions for ExpDropBoxProcRspStruct” on page 479.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**DropBoxName**
The dropoff box name.

- **Max length:** EXP_MAXLEN_SS_DROP_BOX_NAME (20)
- **Min length:** EXP_MINLEN_SS_DROP_BOX_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the dropoff box must exist and must have orders assigned to it.

**ConnectStat**
This is a reserved field; leave it blank. 
Length = 1.
Output arguments for ExpProcDropBox

```c
struct ExpDropBoxProcRspStruct
{
    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_SS_SRCPT_NAME];

    /* The dropoff box name */
    char DropBoxName[EXP_MAXLEN_SS_DROP_BOX_NAME];

    /* The start date of the Information Exchange session */
    char StartDate[EXP_MAXLEN_SS_START_DATE];

    /* The end date of the Information Exchange session */
    char EndDate[EXP_MAXLEN_SS_END_DATE];

    /* The Information Exchange session key (identifier) */
    char Sesskey[EXP_MAXLEN_SS_SESSKEY];

    /* The Information Exchange version being used. */
    char Ieversion[EXP_MAXLEN_SS_IEVERSION];

    /* The Information Exchange release being used */
    char Ierelease[EXP_MAXLEN_SS_IERELEASE];

    /* The status of the last Information Exchange session */
    char LastSess[EXP_MAXLEN_SS_LAST_SESS];

    /* The response code received from Information Exchange */
    char Ierespcode[EXP_MAXLEN_SS_IERESPCODE];

    /* The session end return code */
    char StatusCode[EXP_MAXLEN_SS_STATUS_CODE];
};
```
Output field definitions for ExpDropBoxProcRspStruct

The output can include any or all of these fields.

**SrcptName**
The session receipt name.

Max length: EXP_MAXLEN_SS_SRCPT_NAME (20)
Min length: EXP_MINLEN_SS_SRCPT_NAME (1)
Contents: Alphanumeric data.

**DropBoxName**
The dropoff box name.

Max length: EXP_MAXLEN_SS_DROP_BOX_NAME (20)
Min length: EXP_MINLEN_SS_DROP_BOX_NAME (1)
Contents: Printable data.

**StartDate**
The start date and time of the Information Exchange session.

Max length: EXP_MAXLEN_SS_START_DATE (14)
Min length: EXP_MINLEN_SS_START_DATE (14)
Contents: The date and time in format YYYYMDDHHMMSS.

**EndDate**
The end date and time of the Information Exchange session.

Max length: EXP_MAXLEN_SS_END_DATE (14)
Min length: EXP_MINLEN_SS_END_DATE (14)
Contents: The date and time in format YYYYMDDHHMMSS.

**Sesskey**
The Information Exchange session key.

Max length: EXP_MAXLEN_SS_SESSKEY (8)
Min length: EXP_MINLEN_SS_SESSKEY (8)
Contents: Alphanumeric data.
### Ieversion
The Information Exchange version.

| Max length: | EXP_MAXLEN_SS_IEVERSION (2) |
| Min length: | EXP_MINLEN_SS_IEVERSION (1) |
| Contents:   | Numeric data. |

### Ierelease
The Information Exchange release.

| Max length: | EXP_MAXLEN_SS_IERELEASE (2) |
| Min length: | EXP_MINLEN_SS_IERELEASE (1) |
| Contents:   | Numeric data. |

### LastSess
The status of the last Information Exchange session.

| Max length: | EXP_MAXLEN_SS_LAST_SESS (1) |
| Min length: | EXP_MINLEN_SS_LAST_SESS (1) |
| Contents:   | One of these valid values:  
|             | 0  The last session was successful.  
|             | 1  The last session was not successful. |

### Ierespcode
The response code received from Information Exchange.

| Max length: | EXP_MAXLEN_SS_IERESPCODE (5) |
| Min length: | EXP_MINLEN_SS_IERESPCODE (1) |
| Contents:   | Numeric data. |
Starting and ending Information Exchange sessions

The information in this section describes the functions, the input and output arguments, and the fields used to start and end Information Exchange sessions.

ExpEndSess

This function ends the interactive Information Exchange session. If no session is in progress, the return code is set to 8.

Function prototype for ExpEndSess

```c
int ExpEndSess(
    struct ExpEndSessReqStruct * Input,
    struct ExpSessEndedRspStruct *Output
);
```

Input arguments for ExpEndSess

```
struct ExpEndSessReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The connection status indicator */
    char ConnectStat[EXP_MAXLEN_ES_CONNECT_STAT];

    /* The end session status */
    char SessStat[EXP_MAXLEN_ES_SESS_STAT]

    /* The session start status */
    char StartStatus[EXP_MAXLEN_ES_START_STATUS];

    /* The session end status */
    char EndStatus[EXP_MAXLEN_ES_END_STATUS];

    /* The session start order status */
    char StartOrdrStatus[EXP_MAXLEN_ES_START_ORDR_STATUS];

    /* The session end order status */
    char EndOrdrStatus[EXP_MAXLEN_ES_END_ORDR_STATUS];
};
```
Input field definitions for ExpEndSessReqStruct

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**ConnectStat**
This is a reserved field; leave it blank.
Length = 1.

**SessStat**
This field is not currently in use.

- Max length: EXP_MAXLEN_ES_SESS_STAT (1)
- Min length: EXP_MINLEN_ES_SESS_STAT (1)
- Default: N
- Required: No
- Case sensitive: No
- Validation: No
Output arguments for ExpSessEnded

```c
struct ExpSessEndedRspStruct {
    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_ES_SRCPT_NAME];

    /* The name of the dropoff box for this session */
    char DropBoxName[EXP_MAXLEN_ES_DROP_BOX_NAME];

    /* The starting date of the Information Exchange session */
    char StartDate[EXP_MAXLEN_ES_START_DATE];

    /* The Information Exchange session end date */
    char EndDate[EXP_MAXLEN_ES_END_DATE];

    /* The Information Exchange session key (identifier). */
    char Sesskey[EXP_MAXLEN_ES_SESSKEY];

    /* The Information Exchange version being used. */
    char Ieversion[EXP_MAXLEN_ES_IEVERSION];

    /* The Information Exchange release being used */
    char Ierelease[EXP_MAXLEN_ES_IERELEASE];

    /* The status of the last Information Exchange session */
    char LastSess[EXP_MAXLEN_ES_LAST_SESS];

    /* The response code received from Information Exchange */
    char Ierespcode[EXP_MAXLEN_ES_IERESPCODE];

    /* The status code for session end */
    char StatusCode[EXP_MAXLEN_ES_STATUS_CODE];
};
```
Output field definitions for ExpSessEndedRspStruct

**SrcptName**
The session receipt name.
- Max length: EXP_MAXLEN_ES_SRCPT_NAME (20)
- Min length: EXP_MINLEN_ES_SRCPT_NAME (1)
- Contents: Alphanumeric data.

**DropBoxName**
For interactive sessions, this field is blank.
- Max length: EXP_MAXLEN_ES_DROP_BOX_NAME (20)
- Min length: EXP_MINLEN_ES_DROP_BOX_NAME (1)
- Contents: Blank.

**StartDate**
The time and date the session start command was sent to Information Exchange.
- Max length: EXP_MAXLEN_ES_START_DATE (14)
- Min length: EXP_MINLEN_ES_START_DATE (1)
- Contents: The time and date in YYYYMMDDHHMMSS format.

**EndDate**
The time and date the session end response was received from Information Exchange.
- Max length: EXP_MAXLEN_ES_END_DATE (14)
- Min length: EXP_MINLEN_ES_END_DATE (1)
- Contents: The time and date in YYYYMMDDHHMMSS format.

**Sesskey**
This is the Information Exchange session key.
- Max length: EXP_MAXLEN_SR_SESSKEY (8)
- Min length: EXP_MINLEN_SR_SESSKEY (1)
- Contents: Alphanumeric data.

**Ieversion**
The Information Exchange version being used.
- Max length: EXP_MAXLEN_SR_IEVERSION (2)
- Min length: EXP_MINLEN_SR_IEVERSION (1)
- Contents: Numeric data.
**Ierelease**
The Information Exchange release used.

Max length: EXP_MAXLEN_SR_IERELEASE (2)  
Min length: EXP_MINLEN_SR_IERELEASE (1)  
Contents: Numeric data.

**LastSess**
The status of the last Information Exchange session.

Max length: EXP_MAXLEN_SR_LAST_SESS (5)  
Min length: EXP_MINLEN_SR_LAST_SESS (1)  
Contents: One of these valid values:

0  The last session status is successful.  
1  The last session status is not successful.

**Ierespcode**
The response code received from Information Exchange.

Max length: EXP_MAXLEN_SR_IERESPCODE (5)  
Min length: EXP_MINLEN_SR_IERESPCODE (1)  
Contents: Printable data.

**StatusCode**
The highest severity error that occurred when processing the session. Valid values include:

Max length: EXP_MAXLEN_SR_STATUS_CODE (5)  
Min length: EXP_MINLEN_SR_STATUS_CODE (1)  
Contents: One of these valid values:

0  No errors occurred.  
4  A warning occurred processing one of more orders.  
8  One or more orders had severe errors and could not be processed.  
12  A fatal error occurred and the session was incomplete.  
13  The session is either in progress or was interrupted.
**StartStatus**
The return code from the Information Exchange session start command.

- **Max length:** EXP_MAXLEN_ES_START_STATUS (5)
- **Min length:** EXP_MINLEN_ES_START_STATUS (1)
- **Contents:** Numeric data.

**EndStatus**
The return code from the Information Exchange session end command.

- **Max length:** EXP_MAXLEN_ES_END_STATUS (5)
- **Min length:** EXP_MINLEN_ES_END_STATUS (1)
- **Contents:** Numeric data.

**StartOrdrStatus**
The status for the session start.

- **Max length:** EXP_MAXLEN_ES_START_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_ES_START_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P  Pending
  - I  Issued
  - E  Complete
  - N  Canceled
  - T  Interrupted
  - Z  Incomplete

**EndOrdrStatus**
The status for the session end.

- **Max length:** EXP_MAXLEN_ES_END_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_ES_END_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P  Pending
  - I  Issued
  - E  Complete
  - N  Canceled
  - T  Interrupted
  - Z  Incomplete
ExpStartSess

This function starts interactive dropoff box processing by starting an Information Exchange session. If a session is already in progress, the return code is set to 8.

Only file-level recovery is supported for interactive sessions. When this command completes successfully, your application can process orders interactively.

Function prototype for ExpStartSess

```c
int ExpStartSess(
    struct ExpStartSessReqStruct * Input
    struct ExpSessStartedRspStruct *Output
);
```
**Input arguments for ExpStartSess**

```c
struct ExpStartSessReqStruct
{
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    char IeMboxIdName[EXP_MAXLEN_SS_IE_MBOX_ID_NAME];

    char AppStatus[EXP_MAXLEN_SS_APP_STATUS];

    char Reserved1[1];

    char Reserved2[1];

    char SessRecovery[EXP_MAXLEN_SS_SESS_RECOVERY];

    char Commitdata[EXP_MAXLEN_SS_COMMITDATA];

    char Maxmsgs[EXP_MAXLEN_SS_MAXMSG];

    char Msgsize[EXP_MAXLEN_SS_MSGSIZE];

    char XlateTbl[EXP_MAXLEN_SS_XLATE_TBL];

    char Overwrite[EXP_MAXLEN_SS_OVERWRITE];

    char Check[EXP_MAXLEN_SS_CHECK];

    char TimeZone[EXP_MAXLEN_SS_TIME_ZONE];
};
```
Input field definitions for ExpStartSessReqStruct

For a list of the output fields, see “Output field definitions for ExpSessStartedRspStruct” on page 495.

**ProjName**  
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)  
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)  
- **Required:** Yes  
- **Case sensitive:** No  
- **Validation:** Yes, the project must be open.

**IeMboxIdName**  
The Information Exchange account ID and user ID record name for this session.

- **Max length:** EXP_MAXLEN_SS_IE_MBOX_ID_NAME (10)  
- **Min length:** EXP_MINLEN_SS_IE_MBOX_ID_NAME (1)  
- **Required:** Yes  
- **Case sensitive:** No  
- **Validation:** Yes, the logon record must exist in the Project database.
AppStatus
This indicates whether Expedite for Windows should show the status window during the session.

- Max length: EXP_MAXLEN_SS_APP_STATUS (1)
- Min length: EXP_MINLEN_SS_APP_STATUS (1)
- Default value: RM_DFLT_SS_APP_STATUS (Y)
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - N  Do not display the status window.
  - Y  Display the status window.

Reserved1
This is a reserved field; leave it blank.
Length = 1.

Reserved2
This is a reserved field; leave it blank.
Length = 1.

SessRecovery
The recovery level for the Information Exchange session.

- Max length: EXP_MAXLEN_SS_SESS_RECOVERY (1)
- Min length: EXP_MINLEN_SS_SESS_RECOVERY (1)
- Default value: F
- Required: No
- Case sensitive: Yes
- Validation: The valid value is:
  - F  Use file-level recovery.
**Commitdata**
The number of bytes between commits. This applies only to checkpoint-level recovery. For maximum efficiency, the Commitdata values should be an even multiple of the Msgsize value. Lower values can result in less retransmission of data, if there is a communication failure. Higher values provide faster data transmission.

- **Max length:** EXP_MAXLEN_SS_COMMITDATA (7)
- **Min length:** EXP_MINLEN_SS_COMMITDATA (4)
- **Default value:** 141000
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data between 1000 and 9999999.

**Maxmsgs**
Maximum number of message segments the program can receive between Information Exchange commits. The larger the number specified, the more data Information Exchange sends to you without committing it. A lower number causes more frequent data commits.

- **Max length:** EXP_MAXLEN_SS_MAXMSGSS (3)
- **Min length:** EXP_MINLEN_SS_MAXMSGSS (1)
- **Default value:** 10
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data between 1 and 999.

**Msgsize**
The size of the data segments to send. Your trading partner can take checkpoints only for the message size you specify with this parameter. If you use a large value, your trading partner cannot take frequent checkpoints.

- **Max length:** EXP_MAXLEN_SS_MSGSIZE (5)
- **Min length:** EXP_MINLEN_SS_MSGSIZE (4)
- **Default value:** 47000
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data between 1000 and 999999.
#### XlateTbl

The translation table name to use for all the sends and receives in this session. This value is overridden by the XlateTbl value used on an order. Normally, this should not be changed.

- **Max length:** EXP_MAXLEN_SS_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_SS_XLATE_TBL (1)
- **Default value:** IESTDTBL
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, it must contain the file name of the translate table without the XLT extension.

#### Overwrite

The action to take if a file already exists when receiving data.

- **Max length:** EXP_MAXLEN_SS_OVERWRITE (1)
- **Min length:** EXP_MINLEN_SS_OVERWRITE (1)
- **Default value:** A
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - A Append the new data to the existing file.
  - G Generate a new file name.
  - O Overwrite the existing file.

#### Check

This indicates whether Expedite for Windows should start the session normally or get the status of the previous session. When getting status, Expedite for Windows needs to send a special indicator to Information Exchange so the status of the previous session is preserved, even though it is sending this new session start command.

- **Max length:** EXP_MAXLEN_SS_CHECK (1)
- **Min length:** EXP_MINLEN_SS_CHECK (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Expedite for Windows starts the session normally.
  - Y Expedite for Windows starts the session to get the status of the previous session.
Chapter 5. Writing the interface for Information Exchange

Input field definitions for ExpStartSessReqStruct

TimeZone
The time zone to use to identify your time zone to Information Exchange.

Max length: EXP_MAXLEN_SS_TIME_ZONE (1)
Min length: EXP_MINLEN_SS_TIME-ZONE (1)
Default value: E0000 (Greenwich)
Required: No
Case sensitive: No
Validation: The valid values are:

E1200 New Zealand
E1100 Caroline Islands
E1000 Marian Islands
E1000D Eastern Australia Daylight
E0900 Japan
E0800 China
E0700 Java
E0600 Kazakh
E0500 Pakistan
E0400 Caspian Sea
E0300 Ukraine
E0200 Nile
E0200D Western Europe Daylight
E0100 Europe
E0100D British Summer Time
E0000 Greenwich
W0100 Azores
W0200 Oscar
W0300 Greenland
W0400 Atlantic
W0400D U.S. Eastern Daylight
W0500 U.S. Eastern
W0500D U.S. Central Daylight
W0600 U.S. Central
W0600D U.S. Mountain Daylight
W0700 U.S. Mountain
W0800D U.S. Pacific Daylight
W0800 U.S. Pacific
W0800 Yukon Daylight
W0900 Yukon
W1000 Hawaii
W1100 Bering
Output arguments for ExpSessStarted

```c
struct ExpSessStartedRspStruct
{
    /* The Information Exchange session key (identifier). */
    char Sesskey[EXP_MAXLEN_SS_SESSKEY];

    /* The Information Exchange version used. */
    char Ieversion[EXP_MAXLEN_SS_IEVERSION];

    /* The Information Exchange release used */
    char Ierelease[EXP_MAXLEN_SS_IERELEASE];

    /* The status of the last Information Exchange session */
    char LastSess[EXP_MAXLEN_SS_LAST_SESS];

    /* The response code received from Information Exchange */
    char Ierespcode[EXP_MAXLEN_SS_IERESPCODE];

    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_SS_SRCPT_NAME];

    /* The status of how an order was processed */
    char StatusCode[EXP_MAXLEN_SS_STATUS_CODE];

    /* The status code for the start session command */
    char StartStatus[EXP_MAXLEN_SS_START_STATUS];

    /* The status code for the end session command */
    char EndStatus[EXP_MAXLEN_SS_END_STATUS];

    /* The order status for the start session command */
    char StartOrdrStatus[EXP_MAXLEN_SS_START_ORDR_STATUS];

    /* The order status for the end session command */
    char EndOrdrStatus[EXP_MAXLEN_SS_END_ORDR_STATUS];
};
```
Output field definitions for ExpSessStartedRspStruct

**Sesskey**
The Information Exchange session key.

- Max length: EXP_MAXLEN_SS_SESSKEY (15)
- Min length: EXP_MINLEN_SS_SESSKEY (1)
- Contents: Printable data.

**Ieversion**
The Information Exchange version.

- Max length: EXP_MAXLEN_SS_IEVERSION (2)
- Min length: EXP_MINLEN_SS_IEVERSION (1)
- Contents: Numeric data.

**Ierelease**
The Information Exchange release.

- Max length: EXP_MAXLEN_SS_IERERELEASE (2)
- Min length: EXP_MINLEN_SS_IERERELEASE (1)
- Contents: Numeric data.

**LastSess**
The status of the last Information Exchange session.

- Max length: EXP_MAXLEN_SS_LAST_SESS (5)
- Min length: EXP_MINLEN_SS_LAST_SESS (1)
- Contents: One of these valid values:
  - 0  The last session was successful.
  - 1  The last session was not successful.

**Ierespcode**
The response code received from Information Exchange.

- Max length: EXP_MAXLEN_SS_IERESPCODE (5)
- Min length: EXP_MINLEN_SS_IERESPCODE (1)
- Contents: Numeric data.
**SrcptName**
The session receipt name.

- **Max length:** EXP_MAXLEN_SS_SRCPT_NAME (20)
- **Min length:** EXP_MINLEN_SS_SRCPT_NAME (1)
- **Contents:** Printable data.

**StatusCode**
The highest severity error that occurred when processing the session.

- **Max length:** EXP_MAXLEN_SS_STATUS_CODE (5)
- **Min length:** EXP_MINLEN_SS_STATUS_CODE (5)
- **Contents:** The valid values are:
  - 0  No errors occurred.
  - 4  A warning occurred processing one of more orders.
  - 8  One or more orders had severe errors and could not be processed.
  - 12 A fatal error occurred and the session was incomplete.
  - 13 The session is either in progress or was interrupted.

**StartStatus**
The return code from the Information Exchange session start command.

- **Max length:** EXP_MAXLEN_SS_START_STATUS (5)
- **Min length:** EXP_MINLEN_SS_START_STATUS (1)
- **Contents:** Numeric data.

**EndStatus**
The return code from the Information Exchange session end command.

- **Max length:** EXP_MAXLEN_SS_END_STATUS (5)
- **Min length:** EXP_MINLEN_SS_END_STATUS (1)
- **Contents:** Numeric data.
**StartOrdrStatus**
The status for the session start.

- **Max length:** EXP_MAXLEN_SS_START_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_SS_START_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P  Pending
  - I  Issued
  - E  Complete
  - N  Canceled
  - T  Interrupted
  - Z  Incomplete

**EndOrdrStatus**
The status for the session end.

- **Max length:** EXP_MAXLEN_SS_END_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_SS_END_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P  Pending
  - I  Issued
  - E  Complete
  - N  Canceled
  - T  Interrupted
  - Z  Incomplete
Recovering Information Exchange sessions

The information in this section describes the function, the input and output arguments, and the fields used to recover Information Exchange sessions.

ExpRecoverSessRcpt

This function recovers a session that is in checkpoint restart. In order to recover a session, the status code for the session must be 8, 12, or 13, and the status must be Incomplete or Interrupted. The mailbox ID associated with the session will be disabled. Expedite for Windows must have completed at least one commit. If the return code is 0 and the status is Pending you can resume the session using this function.

If the session is not in need of recovery, the return code will be 350.

Function prototype for ExpRecoverSessRcpt

```c
int ExpRecoverSessRcpt(
    struct ExpRecoverSessRcptReqStruct *Input,
    struct ExpSessRcptRecoveredRspStruct *Output
);
```

Input arguments for ExpRecoverSessRcpt

```c
struct ExpRecoverSessRcptReqStruct
{
    /* Name of the project for the request */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* Name of the session receipt. */
    char SrcptName[EXP_MAXLEN_RC_SRCPT_NAME];

    /* Type of recovery to perform. */
    char Recovery[EXP_MAXLEN_RC_RECOVERY];
};
```
Input field definitions for ExpRecoverSessRcpt

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**SessRcptName**
The name of the session receipt.

- Max length: EXP_MAXLEN_RC_SESS_RCPT_NAME (20)
- Min length: EXP_MINLEN_RC_SESS_RCPT_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the session receipt must exist in the database.

**Recovery**
Indicates the type of recovery to perform.

- Max length: EXP_MAXLEN_RC_RECOVERY (1)
- Min length: EXP_MINLEN_RC_RECOVERY (1)
- Required: Yes
- Case sensitive: Yes
- Validation: The valid values are:
  - **R** Reset the session, marking the unprocessed orders as pending to be resumed at a later time, and enable the mailbox ID. Do not use this option if you were receiving multiple files with a single receive order, or if you were sending multiple EDI envelopes from a single file.
  - **C** Cancel the session. Mark the unprocessed orders as cancelled and enable the mailbox ID.
  - **S** Restart the session from the last checkpoint, or resume a pending session at the beginning of the first order that was not completed.
Output arguments for ExpSessRcptRecovered

struct ExpSessRcptRecoveredRspStruct
{
    /* The session receipt name */
    char SrcptName[EXP_MAXLEN_RC_SRCPT_NAME];

    /* Name of the dropoff box for this session */
    char DropBoxName[EXP_MAXLEN_RC_DROP_BOX_NAME];

    /* The starting date of the Information Exchange session */
    char StartDate[EXP_MAXLEN_RC_START_DATE];

    /* the Information Exchange session end date */
    char EndDate[EXP_MAXLEN_RC_END_DATE];

    /* The Information Exchange session key (identifier). */
    char Sesskey[EXP_MAXLEN_RC_SESSKEY];

    /* The Information Exchange version being used. */
    char Ieversion[EXP_MAXLEN_RC_IEVERSION];

    /* The Information Exchange release being used */
    char Ierelease[EXP_MAXLEN_RC_IERELEASE];

    /* The status of the last Information Exchange session */
    char LastSess[EXP_MAXLEN_RC_LAST_SESS];

    /* The response code received from Information Exchange */
    char Ierespcode[EXP_MAXLEN_RC_IERESPCODE];

    /* The reserved field. */
    char Recovery[EXP_MAXLEN_RC_RECOVERY];

    /* The status code */
    char StatusCode[EXP_MAXLEN_RC_STATUS_CODE];

    /* The session start status */
    char StartStatus[EXP_MAXLEN_RC_START_STATUS];

    /* The session end status */
    char EndStatus[EXP_MAXLEN_RC_END_STATUS];

    /* The session start order status */
    char StartOrdrStatus[EXP_MAXLEN_RC_START_ORDR_STATUS];

    /* The session end order status */
    char EndOrdrStatus[EXP_MAXLEN_RC_END_ORDR_STATUS];
};
Output field definitions for ExpSessRcptRecovered

The output can include any or all of these fields.

**SrcptName**
The session receipt name.

- Max length: EXP_MAXLEN_RC_SRCPT_NAME (20)
- Min length: EXP_MINLEN_RC_SRCPT_NAME (1)
- Contents: Alphanumeric data.

**DropBoxName**
The dropoff box name that was used to process the original session.

- Max length: EXP_MAXLEN_RC_DROP_BOX_NAME (20)
- Min length: EXP_MINLEN_RC_DROP_BOX_NAME (1)
- Contents: Printable data.

**StartDate**
The start date and time of the Information Exchange session.

- Max length: EXP_MAXLEN_RC_START_DATE (14)
- Min length: EXP_MINLEN_RC_START_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**EndDate**
The end date and time of the Information Exchange session.

- Max length: EXP_MAXLEN_RC_END_DATE (14)
- Min length: EXP_MINLEN_RC_END_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**Sesskey**
The Information Exchange session key.

- Max length: EXP_MAXLEN_RC_SESSKEY (8)
- Min length: EXP_MINLEN_RC_SESSKEY (8)
- Contents: Alphanumeric data.
**Ieversion**
The Information Exchange version.

- **Max length:** EXP_MAXLEN_RC_IEVERSION (2)
- **Min length:** EXP_MINLEN_RC_IEVERSION (1)
- **Contents:** Numeric data.

**Ierelease**
The Information Exchange release.

- **Max length:** EXP_MAXLEN_RC_IERELEASE (2)
- **Min length:** EXP_MINLEN_RC_IERELEASE (1)
- **Contents:** Numeric data.

**LastSess**
The status of the last Information Exchange session.

- **Max length:** EXP_MAXLEN_RC_LAST_SESS (1)
- **Min length:** EXP_MINLEN_RC_LAST_SESS (1)
- **Contents:**
  - One of these valid values:
    - 0 The last session was successful.
    - 1 The last session was not successful.

**Ierespcode**
The response code received from Information Exchange.

- **Max length:** EXP_MAXLEN_RC_IERESPCODE (5)
- **Min length:** EXP_MINLEN_RC_IERESPCODE (1)
- **Contents:** Numeric data.

**Recovery**
This is a reserved field, length = 1.

**Statuscode**
The session status code.

- **Max length:** EXP_MAXLEN_RC_STATUS_CODE (5)
- **Min length:** EXP_MINLEN_RC_STATUS_CODE (1)
- **Contents:** Numeric data.
Chapter 5. Writing the interface for Information Exchange

Output field definitions for ExpSessRcptRecovered

**StartStatus**
The return code from the Information Exchange session start command.

- **Max length:** EXP_MAXLEN_RC_START_STATUS (5)
- **Min length:** EXP_MINLEN_RC_START_STATUS (1)
- **Contents:** Numeric data.

**EndStatus**
The return code from the Information Exchange session end command.

- **Max length:** EXP_MAXLEN_RC_END_STATUS (5)
- **Min length:** EXP_MINLEN_RC_END_STATUS (1)
- **Contents:** Numeric data.

**StartOrdrStatus**
The status for the session start.

- **Max length:** EXP_MAXLEN_RC_START_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_RC_START_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P Pending
  - I Issued
  - E Complete
  - N Canceled
  - T Interrupted
  - Z Incomplete

**EndOrdrStatus**
The status for the session end.

- **Max length:** EXP_MAXLEN_RC_END_ORDR_STATUS (1)
- **Min length:** EXP_MINLEN_RC_END_ORDR_STATUS (1)
- **Contents:** The valid values are:
  - P Pending
  - I Issued
  - E Complete
  - N Canceled
  - T Interrupted
  - Z Incomplete
Interactive session commands

This section lists the functions, the input and output arguments, and the field descriptions you use to interact with Information Exchange.

ExpDoPurgOrdr

Use this function to delete a file from your Information Exchange Mailbox. You must specify the message key for the file, which is available from the ExpDoQuryOrdr response or from Information Exchange Administration Services. If the file with the specified message key does not exist in the mailbox, or if the file is being received, or if the Information Exchange user profile Use purge message command field is set to N, the return code will be set to 4 and no file will be deleted.

Function prototype for ExpDoPurgOrdr

```c
int ExpDoPurgOrdr(
    struct ExpDoPurgOrdrReqStruct * Input,
    struct ExpPurgOrdrDoneRspStruct * Output
);
```

Input arguments for ExpDoPurgOrdr

```c
struct ExpDoPurgOrdrReqStruct
{
    /* The project name for the request */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The message key */
    char Msgkey[EXP_MAXLEN_DP_MSGKEY];
};
```
Input field definitions for ExpDoPurgOrdrReqStruct

For a list of the output fields, see “Output field definitions for ExpPurgOrdrDoneRspStruct” on page 506.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**Msgkey**
The message key for the file to be deleted.

- **Max length:** EXP_MAXLEN_DP_MSGKEY (20)
- **Min length:** EXP_MINLEN_DP_MSGKEY (20)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** None.
Output arguments for ExpDoPurgOrdr

```c
struct ExpPurgOrdrDoneRspStruct
{
    /* The purge order return code */
    StatusCode[EXP_MAXLEN_DP_STATUS_CODE];
};
```

Output field definitions for `ExpPurgOrdrDoneRspStruct`

The output can include the following:

**StatusCode**
Indicates the status code for the purge order processing.

- **Max length:** EXP_MAXLEN_SR_STATUS_CODE (5)
- **Min length:** EXP_MINLEN_SR_STATUS_CODE (1)
- **Contents:** Numeric characters.
ExpDoQuryOrdr

Use this function to get a list of items in the Information Exchange mailbox. If there are no items in the mailbox, the return code will be set to 4.

The ExpDoQuryOrdr function works the same as other Expedite “list” functions. Issue the command for the first time with StartIndex set to 1. You must specify the number of items that Expedite for Windows should include in the response in the NumItems field. For subsequent calls, set StartIndex to one more than the previous StartIndex plus the NumItems received to get the next group of items. When Expedite for Windows sets NumItems to negative one (-1), there are no more items to list for the query response.

Function prototype for ExpDoQuryOrdr

```c
int ExpDoQuryOrdr(  
    struct ExpListItemsReqStruct *Input,  
    unsigned short *EntriesReturned,  
    struct ExpQuryOrdrDoneRspStruct Output[]  
);
```

Input arguments for ExpDoQuryOrdr

```c
struct ExpListItemsReqStruct  
{  
    /* The project name */  
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The number of entries requested for receiving */  
    char NumEntries[EXP_MAXLEN_CF_NUM_ENTRIES];

    /* The entry number at which to start the search */  
    char StartIndex[EXP_MAXLEN_CF_START_INDEX];
};
```
Input field definitions for ExpDoQuryOrdrReqStruct

For a list of the output fields, see “Output field definitions for ExpQuryOrdrDoneRspStruct” on page 511.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**NumEntries**
The number of entries to return in the response structure.

- **Max length:** EXP_MAXLEN_CF_NUM_ENTRIES (5)
- **Min length:** EXP_MINLEN_NUM_ENTRIES (1)
- **Max value:** EXP_MAXVAL_NUM_ENTRIES (65535)
- **Min value:** EXP_MINVAL_NUM_ENTRIES (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**StartIndex**
The entry number at which to start the search.

- **Max length:** EXP_MAXLEN_CF_START_INDEX (5)
- **Min length:** EXP_MINLEN_START_INDEX (1)
- **Max value:** EXP_MAXVAL_START_INDEX (65535)
- **Min value:** EXP_MINVAL_START_INDEX (1)
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**EntriesReturned**
The number of items copied into the output structure.
Output arguments for ExpDoQuryOrdr

struct ExpQuryOrdrDoneRspStruct
{
    /* The status of how an order was processed */
    char StatusCode[EXP_MAXLEN_XQ_STATUS_CODE];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_XQ_IE_SYSID];

    /* The Information Exchange account ID */
    char IeMboxId[EXP_MAXLEN_XQ_IE_MBOX_ID];

    /* The message key */
    char Msgkey[EXP_MAXLEN_XQ_MSGKEY];

    /* The Information Exchange message class */
    char MsgClass[EXP_MAXLEN_XQ_MSG_CLASS];

    /* The message name */
    charMsgName[EXP_MAXLEN_XQ_MSG_NAME];

    /* The message sequence number */
    char MsgSeqNo[EXP_MAXLEN_XQ_MSG_SEQ_NO];

    /* A random ID assigned to the data */
    char Uniqueid[EXP_MAXLEN_XQ_UNIQUEID];

    /* The date the received data went to Information Exchange */
    char MsgDate[EXP_MAXLEN_XQ_MSG_DATE];

    /* The length of the file received */
    char Length[EXP_MAXLEN_XQ_LENGTH];

    /* Whether the data is text, binary, or EDI */
    char Datatype[EXP_MAXLEN_XQ_DATATYPE];

    /* The type of EDI data received */
    char Editype[EXP_MAXLEN_XQ_EDITYPE];

    /* The name of the system that sent the data */
    char Sysname[EXP_MAXLEN_XQ_SYSNAME];

    /* The level of the system that sent the data */
    char Syslevel[EXP_MAXLEN_XQ_SYSLEVEL];

    /* The file name on the sender's system */
    char Senderfile[EXP_MAXLEN_XQ_SENDERFILE];

    /* The directory where the file was stored */
    char Senderloc[EXP_MAXLEN_XQ_SENDERLOC];
}
/* A free-format description of the data */
char Description[EXP_MAXLEN_XQ_DESCRIPTION];

{/* The date of the file on sender's system */
char Filedate[EXP_MAXLEN_XQ_FILEDATE];

{/* The record format of the file on the sender's system */
char Recfm[EXP_MAXLEN_XQ_RECFM];

{/* The original record length of the file */
char Reclen[EXP_MAXLEN_XQ_RECLEN];

{/* The method used to delimit the records */
char Recdlm[EXP_MAXLEN_XQ_RECDLM];

{/* The type of system that sent the data */
char Systype[EXP_MAXLEN_XQ_SYSTYPE];

{/* The software version of the system */
char Sysver[EXP_MAXLEN_XQ_SYSVER];

{/* The name of the software compression */
char Comsw[EXP_MAXLEN_XQ_COMSW];

{/* The version of the software compression */
char Comver[EXP_MAXLEN_XQ_COMVER];

{/* The name of the compressed file */
char Comfile[EXP_MAXLEN_XQ_COMFILE];

{/* The data class */
char Mode[EXP_MAXLEN_XQ_MODE];
};
Output field definitions for ExpQuryOrdrDoneRspStruct

The output can include any or all of these fields.

**StatusCode**
Indicates a return code for the query order processing.

- **Max length:** EXP_MAXLEN_XQ_STATUS_CODE (5)
- **Min length:** EXP_MINLEN_XQ_STATUS_CODE (1)
- **Contents:** Numeric data.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. The IeSysid is specified with the IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_XQ_IE_SYSID (3)
- **Min length:** EXP_MINLEN_XQ_IE_SYSID (1)
- **Contents:** Printable data.

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight characters are the user ID.

- **Max length:** EXP_MAXLEN_XQ_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_XQ_IE_MBOX_ID (1)
- **Contents:** Printable data.

**MsgKey**
The unique identifier assigned by Information Exchange that can be used to retrieve a specific file from the mailbox.

- **Max length:** EXP_MAXLEN_XQ_MSG_KEY (20)
- **Min length:** EXP_MINLEN_XQ_MSG_KEY (20)
- **Contents:** Alphanumeric characters.
Output field definitions for ExpQuryOrdrDoneRspStruct

**MsgClass**
The Information Exchange user message class specified by the sender.

Max length: EXP_MAXLEN_XQ_MSG_CLASS (8)
Min length: EXP_MINLEN_XQ_MSG_CLASS (1)
Contents: Printable data.

**MsgName**
The message name specified by the sender.

Max length: EXP_MAXLEN_XQ_MSG_NAME (8)
Min length: EXP_MINLEN_XQ_MSG_NAME (1)
Contents: Printable data.

**MsgSeqNo**
The message sequence number specified by the sender.

Max length: EXP_MAXLEN_XQ_MSG_SEQ_NO (6)
Min length: EXP_MINLEN_XQ_MSG_SEQ_NO (1)
Contents: Printable data.

**Uniqueid**
The ID assigned to the data by the sending interface. Use it to identify and associate the data with any acknowledgments received.

Max length: EXP_MAXLEN_XQ_UNIQUEID (8)
Min length: EXP_MINLEN_XQ_UNIQUEID (1)
Contents: Printable data.

**MsgDate**
The date the data was received and placed into the Information Exchange mailbox.

Max length: EXP_MAXLEN_XQ_MSG_DATE (14)
Min length: EXP_MINLEN_XQ_MSG_DATE (14)
Contents: The date and time in YYYYMMDDHHMMSS format.
Chapter 5. Writing the interface for Information Exchange

Output field definitions for ExpQuryOrdrDoneRspStruct

**Length**
The length (as it is in the Information Exchange mailbox) of the file received.

- **Max length:** EXP_MAXLEN_XQ_LENGTH (9)
- **Min length:** EXP_MINLEN_XQ_LENGTH (1)
- **Contents:** Numeric data.

**Datatype**
This indicates whether the data is text, binary, or EDI.

- **Max length:** EXP_MAXLEN_XQ_DATATYPE (1)
- **Min length:** EXP_MINLEN_XQ_DATATYPE (1)
- **Contents:** One of these valid values:
  - B Binary data.
  - E Supported EDI formatted data.
  - T Text data.

**Editype**
The type of EDI data received: X12, UCS, UN/TDI, EDIFACT, or unformatted.

- **Max length:** EXP_MAXLEN_XQ_EDITYPE (1)
- **Min length:** EXP_MINLEN_XQ_EDITYPE (1)
- **Contents:** X12, UCS, UN/TDI, EDIFACT, or unformatted data.

**Sysname**
The name of the system that sent the data.

- **Max length:** EXP_MAXLEN_XQ_SYSNAME (8)
- **Min length:** EXP_MINLEN_XQ_SYSNAME (1)
- **Contents:** Printable data.

**Syslevel**
The system level of the system that sent the data.

- **Max length:** EXP_MAXLEN_XQ_SYSLEVEL (8)
- **Min length:** EXP_MINLEN_XQ_SYSLEVEL (1)
- **Contents:** Printable data.
Senderfile
The original file name on the sender's system.

Max length: EXP_MAXLEN_XQ_SENDERFILE (54)
Min length: EXP_MINLEN_XQ_SENDERFILE (1)
Contents: A file name.

Senderloc
If the sending system was a workstation, the directory where the file was stored on the sender's system.

Max length: EXP_MAXLEN_XQ_SENDERLOC (65)
Min length: EXP_MINLEN_XQ_SENDERLOC (1)
Contents: A file path.

Description
The free-format data description.

Max length: EXP_MAXLEN_XQ_DESCRIPTION (79)
Min length: EXP_MINLEN_XQ_DESCRIPTION (1)
Contents: As entered by the sender.

Filedate
The date the file was created or last edited on the sender's system.

Max length: EXP_MAXLEN_XQ_FILEDATE (14)
Min length: EXP_MINLEN_XQ_FILEDATE (14)
Contents: The date and time in format YYYYMMDDHHMMSS.

Recfm
The record format of the file on the sender's system. If record format is not used by the sender’s system, then four question marks (????) will be specified.

Max length: EXP_MAXLEN_XQ_RECFM (4)
Min length: EXP_MINLEN_XQ_RECFM (1)
Contents: Printable data.
### Reclen
The file’s original record length on the sender’s system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Max length</th>
<th>Min length</th>
<th>Max value</th>
<th>Min value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reclen</td>
<td>EXP_MAXLEN_XQ_RECLLEN (5)</td>
<td>EXP_MINLEN_SN_RECLLEN (1)</td>
<td>EXP_MAXVAL_SN_RECORDLEN (65535)</td>
<td>EXP_MINVAL_SN_RECORDLEN (1)</td>
<td>Numeric data</td>
</tr>
</tbody>
</table>

### Recdlm
The method used to delimit the records.

<table>
<thead>
<tr>
<th>Field</th>
<th>Max length</th>
<th>Min length</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recdlm</td>
<td>EXP_MAXLEN_XQ_RECDLM (1)</td>
<td>EXP_MINLEN_XQ_RECDLM (1)</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C CRLF delimiters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>E EDI characters delimit the data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L 2-byte length precedes each record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N Either the records have no delimiters or the CDH does not indicate the type of delimiter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U Unknown delimiters. Expedite for Windows will not process delimiters when the data is received.</td>
</tr>
</tbody>
</table>

### Delimited
This indicates whether the data is delimited by carriage-return, line-feed characters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Max length</th>
<th>Min length</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimited</td>
<td>EXP_MAXLEN_XQ_DELIMITED (1)</td>
<td>EXP_MINLEN_XQ_DELIMITED (1)</td>
<td>One of these valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N Data is not delimited by CRLF characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Y Data is delimited by CRLF characters.</td>
</tr>
</tbody>
</table>
**Systype**
The type of system that sent the data.

- **Max length:** EXP_MAXLEN_XQ_SYSTYPE (2)
- **Min length:** EXP_MINLEN_XQ_SYSTYPE (2)
- **Contents:** Printable data.
- **System type:** One of these valid values:
  - 01 Unknown system type
  - 10 expEDIte/PC
  - 11 Expedite Base/2
  - 12 Expedite Base/AIX
  - 14 Expedite Base for SCO UNIX
  - 15 Expedite Base/DOS
  - 16 Expedite Base for SCO XENIX
  - 17 Expedite Base for Windows
  - 19 Expedite for Windows
  - 20 expEDIte/MVS Host
  - 21 Expedite Base/MVS
  - 22 TCP/IP FTP Gateway
  - 30 Mail Exchange
  - 31 Expedite Base/VM
  - 33 X.400 Gateway
  - 40 Expedite/Direct
  - 44 EDI VAN Interconnect
  - 60 MQSeries Services
  - 61 EDI Server
  - 71 Expedite Base/400
  - 80 Expedite/CICS
  - 90 Information Exchange Administration Services
  - 91 Expedite/Async

**Sysver**
The software version of the Expedite for Windows system sending the data.

- **Max length:** EXP_MAXLEN_XQ_SYSVER (3)
- **Min length:** EXP_MINLEN_XQ_SYSVER (1)
- **Contents:** Printable data.
Chapter 5. Writing the interface for Information Exchange

Output field definitions for ExpQuryOrdrDoneRspStruct

Comsw
The compression software package name used to compress the file.

- Max length: EXP_MAXLEN_XQ_COMSW (10)
- Min length: EXP_MINLEN_XQ_COMSW (1)
- Contents: Printable data.

Comver
The version of the software compression package used to compress the file.

- Max length: EXP_MAXLEN_XQ_COMVER (5)
- Min length: EXP_MINLEN_XQ_COMVER (1)
- Contents: Printable data.

Comfile
The name of the compressed file.

- Max length: EXP_MAXLEN_XQ_COMFILE (54)
- Min length: EXP_MINLEN_XQ_COMFILE (1)
- Contents: A file name.

Mode
This indicates a test or a normal file.

- Max length: EXP_MAXLEN_XQ_MODE (1)
- Min length: EXP_MINLEN_XQ_MODE (1)
- Contents: One of these valid values:
  - N  Normal mode.
  - T  Test mode.
ExpDoRecvOrdr

This function processes a receive order interactively. Control returns to your application when the order is processed. The output is an array of structures that correspond to the RECEIVED record in Expedite Base.

More than one file can be received for each receive order. Issue the command with NumRcpts set to the number of receipts the output-array, memory area can accommodate. Your application must issue the command repeatedly, until RcptsRead is set to 0 in order to retrieve all the order receipts generated by this receive order. The order processes only once.

Function prototype for ExpDoRecvOrdr

```c
int ExpDoRecvOrdr(
    struct ExpDoRecvOrdrReqStruct * Input,
    struct ExpRecvOrdrDoneRspStruct * Output
);
```
Input arguments for ExpDoRecvOrdr

```
struct ExpDoRecvOrdrReqStruct {
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_XR_DV_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_DV_IE_MBOX_ID];

    /* The table type of G, O, or P */
    char TableType[EXP_MAXLEN_DV_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_DV_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_DV_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_DV_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_DV_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_DV_ADDR_TYPE];

    /* The user message class */
    char MsgClass[EXP_MAXLEN_DV_MSG_CLASS];

    /* The file name for the received data */
    char Fileid[EXP_MAXLEN_DV_FILEID];

    /* The archive ID of received data */
    char Archiveid[EXP_MAXLEN_DV_ARCHIVEID];

    /* The option to receive one file or all files that match */
    char Allfiles[EXP_MAXLEN_DV_ALLFILES];

    /* The option to receive multiple files */
    char Multfiles[EXP_MAXLEN_DV_MULTFILES];

    /* The option to receive files with original names */
    char Origfile[EXP_MAXLEN_DV_ORGFILE];
```

/* The format for the received data */
char Format[EXP_MAXLEN_DV_FORMAT];

/* The EDI record size */
char Recordsize[EXP_MAXLEN_DV_RECORDSIZE];

/* The processing for line-length delimiters */
char Processlen[EXP_MAXLEN_DV_PROCESSLEN];

/* The automatic processing for EDI data */
char Autoedi[EXP_MAXLEN_DV_AUTOEDI];

/* The options for EDI data processing */
char Ediopt[EXP_MAXLEN_DV_EDIOPT];

/* Reserved field */
char Reserved[20];

/* The option to receive requeued files */
char Requeued[EXP_MAXLEN_DV_REQUEUED];

/* The option to remove EOF characters */
char Removeeof[EXP_MAXLEN_DV_REMOVEEOF];

/* The option to receive only non-EDI data */
char Nonedionly[EXP_MAXLEN_DV_NONEDIONLY];

/* The option to receive only EDI data */
char Edionly[EXP_MAXLEN_DV_EDIONLY];

/* The option to overwrite, append, or generate a new file name */
char Overwrite[EXP_MAXLEN_DV_OVERWRITE];

/* The wait time to receive a response */
char Waittime[EXP_MAXLEN_DV_WAITTIME];

/* The message key */
char Msgkey[EXP_MAXLEN_DV_MSGKEY];

/* The session start date and time */
char StartDate[EXP_MAXLEN_DV_START_DATE];

/* The session end date and time */
char EndDate[EXP_MAXLEN_DV_END_DATE];

/* The time zone */
char GlTimeZone[EXP_MAXLEN_DV_GL_TIMEZONE];

/* The translation table */
char XlateTbl[EXP_MAXLEN_DV_XLATE_TBL];

/* This indicates the receive is for EDI data */
char Receiveedi[EXP_MAXLEN_DV_RECEIVEEDI];
};
Input field definitions for ExpDoRecvOrdrReqStruct

For a list of output fields, see “Output field definitions for ExpRecvOrdrDoneRspStruct” on page 535.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- Max length: EXP_MAXLEN_PJ_PROJ_NAME (20)
- Min length: EXP_MINLEN_PJ_PROJ_NAME (1)
- Required: Yes
- Case sensitive: No
- Validation: Yes, the project must be open.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. If you specify IeSysid, you must specify IeMboxId to create the Information Exchange address.

- Max length: EXP_MAXLEN_DV_IE_SYSID (3)
- Min length: EXP_MINLEN_DV_IE_SYSID (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with 8 fixed characters and padded on the right with blanks, if needed. The first 8 characters are the account ID; the next 8 are the user ID.

- Max length: EXP_MAXLEN_DV_IE_MBOX_ID (16)
- Min length: EXP_MINLEN_DV_IE_MBOX_ID (9)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
**TableType**
The alias table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- Max length: `EXP_MAXLEN_DV_TABLE_TYPE` (1)
- Min length: `EXP_MINLEN_DV_TABLE_TYPE` (1)
- Default value: Blank
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - G Alias table for global account.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The alias table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

- Max length: `EXP_MAXLEN_DV_ALIAS_TBL` (3)
- Min length: `EXP_MINLEN_DV_ALIAS_TBL` (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

**Aliasname**
This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

- Max length: `EXP_MAXLEN_DV_ALIASNAME` (16)
- Min length: `EXP_MINLEN_DV_ALIASNAME` (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No
Listname
The name of the list.

- Max length: EXP_MAXLEN_DV_LISTNAME (8)
- Min length: EXP_MINLEN_DV_LISTNAME (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

Nickname
This is a nickname from the address book. Use the Nickname with the AddrType to determine which network address to use to send or receive data. If you specify Nickname, you must specify AddrType.

- Max length: EXP_MAXLEN_DV_NICKNAME (20)
- Min length: EXP_MINLEN_DV_NICKNAME (1)
- Default value: Blank
- Required: No
- Case sensitive: No
- Validation: No

AddrType
This is the address type to use to send or receive data. Use the AddrType with the Nickname to determine which network address to use to send or receive data. If you specify AddrType, you must specify Nickname.

- Max length: EXP_MAXLEN_DV_ADDR_TYPE (1)
- Min length: EXP_MINLEN_DV_ADDR_TYPE (1)
- Default value: E
- Required: No
- Case sensitive: Yes
- Validation: The valid values are:
  - A Use the alias name and the alias table.
  - D Use a distribution list with the name specified in the Nickname field.
  - E Use the Information Exchange address.
  - X Use the Mail Exchange address.
**Input field definitions for ExpDoRecvOrdrReqStruct**

**MsgClass**
The Information Exchange user message class specified by the sender.

- **Max length:** EXP_MAXLEN_DV_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_DV_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Fileid**
This is the name of the file where the received data will be stored.

- **Max length:** EXP_MAXLEN_DV_FILEID (65)
- **Min length:** EXP_MINLEN_DV_FILEID (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** Yes, as a file path.

**Archiveid**
This is the archive reference identifier that Expedite for Windows sends to Information Exchange for received files.

- **Max length:** EXP_MAXLEN_DV_ARCHIVEID (8)
- **Min length:** EXP_MINLEN_DV_ARCHIVEID (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Allfiles**
The option to receive the first file or all files in the Information Exchange mailbox that match the receive specifications.

- **Max length:** EXP_MAXLEN_DV_ALLFILES (1)
- **Min length:** EXP_MINLEN_DV_ALLFILES (1)
- **Default value:** Y
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Receive first matching file.
  - Y  Receive all matching files.

**Multfiles**
This option determines whether received files are put into separate files or concatenated into one file.

- **Max length:** EXP_MAXLEN_DV_MULTFILES (1)
- **Min length:** EXP_MINLEN_DV_MULTFILES (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Store the data in a single file.
  - Y  Store the files received in individual files. Use the file mask specified to generate new file names, if necessary.
**Origfile**
This option indicates whether Expedite for Windows should receive data into a file using the original name from the sending system.

- **Max length:** EXP_MAXLEN_DV_ORIGFILE (1)
- **Min length:** EXP_MINLEN_DV_ORIGFILE (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Do not use the original file name.
  - Y  Use the original file name from the sending system if it is a valid file name for the current system. If it is not, the OVERWRITE option will be considered.

**Format**
The option for Expedite for Windows to format the received file as e-mail.

- **Max length:** EXP_MAXLEN_DV_FORMAT (1)
- **Min length:** EXP_MINLEN_DV_FORMAT (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Do not format as e-mail.
  - Y  Format the text data as 79-byte records, padded with blanks. The default user class will be FFMSG001.
Recordsize
This indicates whether the EDI data is broken up into fixed-length records by placing CRLF characters in the data at specified intervals.

- **Max length:** EXP_MAXLEN_DV_RECORDSIZE (3)
- **Min length:** EXP_MINLEN_DV_RECORDSIZE (1)
- **Max value:** EXP_MAXVAL_DV_RECORDSIZE (999)
- **Min value:** EXP_MINVAL_DV_RECORDSIZE (0)
- **Default value:** 0
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

Processlen
This option controls the processing of length delimiters at the beginning of each record.

- **Max length:** EXP_MAXLEN_DV_PROCESSLEN (1)
- **Min length:** EXP_MINLEN_DV_PROCESSLEN (1)
- **Default value:** C
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - C  Convert the line length delimiters to CRLF delimiters.
  - I  Ignore line length delimiters.
  - R  Remove line length delimiters.
**Autoedi**
This option indicates whether EDI processing for files is done automatically, if the CDH indicates that the files are EDI formatted.

- **Max length:** EXP_MAXLEN_DV_AUTOEDI (1)
- **Min length:** EXP_MINLEN_DV_AUTOEDI (1)
- **Default value:** Y
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F Ignore the CDH and process the data as EDI. If the data is not EDI, Expedite for Windows receives it as is and writes a warning on the receive receipt for this file.
  - N Do not process the data as EDI.
  - Y Automatically process the data as EDI.

**Ediopt**
This option indicates whether CRLF characters are added after each segment delimiter in the EDI file.

- **Max length:** EXP_MAXLEN_DV_EDIOPT (1)
- **Min length:** EXP_MINLEN_DV_EDIOPT (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Do not add CRLF at the end of EDI segments.
  - Y Add CRLF characters at the end of EDI segments.
  - B Add CRLF after bytes specified in Recordsize.

**Reserved1**
This field is reserved; leave it blank.
Length = 20.
Requeued
This option indicates whether requeued files are received. If you specify Y for Requeued, do not specify a source address (nickname, account, user ID, system ID, Mail Exchange IEA, and so on).

- **Max length:** EXP_MAXLEN_DV_REQUEUED (1)
- **Min length:** EXP_MINLEN_DV_REQUEUED (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Do not receive requeued files.
  - Y Receive requeued files. Do not specify a destination on the RECEIVE command if you specify this option.

Removeeof
This option indicates whether EOF characters are removed from the end of a received file.

- **Max length:** EXP_MAXLEN_DV_REMOVEEOF (1)
- **Min length:** EXP_MINLEN_DV_REMOVEEOF (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Do not remove EOF characters.
  - Y Remove EOF characters.
**Nonedionly**
This option lets you only receive data that is not EDI data per the CDH. Nonedionly and Edionly cannot both be set to Y.

- **Max length:** EXP_MAXLEN_DV_NONEDIONLY (1)
- **Min length:** EXP_MINLEN_DV_NONEDIONLY (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Receive all data.
  - Y  Check the CDH and receive only non-EDI data.

**Edionly**
This option lets you only receive data that is in EDI format per the CDH. Nonedionly and Edionly cannot both be set to Y.

- **Max length:** EXP_MAXLEN_DV_EDIONLY (1)
- **Min length:** EXP_MINLEN_DV_EDIONLY (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N  Receive all data.
  - Y  Check the CDH and receive only EDI data.

**Overwrite**
This option indicates the action to take if a file already exists when receiving data.

- **Max length:** EXP_MAXLEN_DV_OVERWRITE (1)
- **Min length:** EXP_MINLEN_DV_OVERWRITE (1)
- **Default value:** A
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - A  Append the new data to the existing file.
  - G  Generate a new file name using the file mask specified for the new data.
  - O  Overwrite the existing file.
Chapter 5. Writing the interface for Information Exchange

Input field definitions for ExpDoRecvOrdReqStruct

**Waittime**
This indicates the number of minutes to wait for a response to a receive command from Information Exchange.

- **Max length:** EXP_MAXLEN_DV_WAITTIME (2)
- **Min length:** EXP_MINLEN_DV_WAITTIME (1)
- **Max value:** EXP_MAXVAL_DV_WAITTIME (5)
- **Min value:** EXP_MINVAL_DV_WAITTIME (0)
- **Default value:** 0
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**Msgkey**
This indicates a unique message key you can use to receive a specific file from a mailbox.

- **Max length:** EXP_MAXLEN_DV_MSGKEY (20)
- **Min length:** EXP_MINLEN_DV_MSGKEY (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**StartDate**
This is the start date and time that Expedite for Windows should receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_DV_START_DATE (14)
- **Min length:** EXP_MINLEN_DV_START_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** The date and time in format YYYYMMDDHHMMSS.
**EndDate**
This is the end date and time that Expedite for Windows should receive files from Information Exchange.

- **Max length:** EXP_MAXLEN_DV_END_DATE (14)
- **Min length:** EXP_MINLEN_DV_END_DATE (14)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** The date and time in format YYYYMMDDHHMMSS.

**GlTimeZone**
Indicates whether the date and time field specified on the receive order is local time (as specified in the TimeZone field on the dropoff box or **ExpStartSess** function) or Greenwich mean time.

- **Max length:** EXP_MAXLEN_DV_GL_TIMEZONE (1)
- **Min length:** EXP_MINLEN_DV_GL_TIMEZONE (1)
- **Default value:** L
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  - **G**  Greenwich mean time (GMT)
  - **L**  Local time.

**XlateTbl**
This is the translation table name.

- **Max length:** EXP_MAXLEN_DV_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_DV_XLATE_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, it must contain the file name of the translation table without the XLT extension.
**Receiveedi**

The option to receive data as EDI or unformatted.

| **Max length:** | EXP_MAXLEN_DV_RECEIVEEDI (1) |
| **Min length:** | EXP_MINLEN_DV_RECEIVEEDI (1) |
| **Default value:** | N |
| **Required:** | No |
| **Case sensitive:** | Yes |
| **Validation:** | The valid values are: |
| | N  Receive as unformatted data. |
| | Y  Receive as EDI data. |
Output arguments for ExpDoRecvOrdr

```
struct ExpRecvOrdrDoneRspStruct
{
    /* The receive receipt name */
    char RcptName[EXP_MAXLEN_XR_RCPT_NAME];

    /* The status of the processed order*/
    char StatusCode[EXP_MAXLEN_XR_STATUS_CODE];

    /* The Expedite for Windows session start date */
    char PstartDate[EXP_MAXLEN_XR_PSTART_DATE];

    /* The Expedite for Windows session end date */
    char PendDate[EXP_MAXLEN_XR_PEND_DATE];

    /* The number of files received */
    char FilesRcvd[EXP_MAXLEN_XR_FILES_RCVD];
};
```
Chapter 5. Writing the interface for Information Exchange

Output field definitions for ExpRecvOrdrDoneRspStruct

The output can include any or all of these fields.

**ReptName**
The receive receipt name.

- Max length: EXP_MAXLEN_XR_RCPT_NAME (20)
- Min length: EXP_MINLEN_XR_RCPT_NAME (1)
- Case sensitive: No
- Contents: Printable data.

**StatusCode**
The status of a processed order.

- Max length: EXP_MAXLEN_XR_STATUS_CODE (5)
- Min length: EXP_MINLEN_XR_STATUS_CODE (1)
- Contents: Numeric data.

**PstartDate**
The processing start date and time.

- Max length: EXP_MAXLEN_XR_PSTART_DATE (14)
- Min length: EXP_MINLEN_XR_PSTART_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**PendDate**
The processing end date and time.

- Max length: EXP_MAXLEN_XR_PEND_DATE (14)
- Min length: EXP_MINLEN_XR_PEND_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**FilesRcvd**
The number of files received as a result of processing this order.

- Max length: EXP_MAXLEN_XR_FILES_RCVD (5)
- Min length: EXP_MINLEN_XR_FILES_RCVD (1)
- Contents: Numeric data.
ExpDoSendOrdr

Use this function to process a send order in an interactive session. Control returns to your application as soon as the order is processed. The output structure is like the SENT record in Expedite Base.

Function prototype for ExpDoSendOrdr

```c
int ExpDoSendOrdr(
    struct ExpDoSendOrdrReqStruct * Input,
    struct ExpSendOrdrDoneRspStruct * Output
);
```

Input arguments to ExpDoSendOrdr

```c
struct ExpDoSendOrdrReqStruct
{
    /* The project name */
    char ProjName[EXP_MAXLEN_PJ_PROJ_NAME];

    /* The Information Exchange system ID */
    char IeSysid[EXP_MAXLEN_DS_IE_SYSID];

    /* The Information Exchange account ID and user ID */
    char IeMboxId[EXP_MAXLEN_DS_IE_MBOX_ID];

    /* The table type G, O, or P */
    char TableType[EXP_MAXLEN_DS_TABLE_TYPE];

    /* The Information Exchange alias table */
    char AliasTbl[EXP_MAXLEN_DS_ALIAS_TBL];

    /* The Information Exchange alias name */
    char Aliasname[EXP_MAXLEN_DS_ALIASNAME];

    /* The list name */
    char Listname[EXP_MAXLEN_DS_LISTNAME];

    /* The nickname */
    char Nickname[EXP_MAXLEN_DS_NICKNAME];

    /* The address type */
    char AddrType[EXP_MAXLEN_DS_ADDR_TYPE];
};
```
/* The Information Exchange user message class */
char MsgClass[EXP_MAXLEN_DS_MSG_CLASS];

/* The file name to send */
char Fileid[EXP_MAXLEN_DS_FILEID];

/* The option to verify the receiver’s address */
char Verify[EXP_MAXLEN_DS_VERIFY];

/* The option to proceed if address verification fails */
char EdiVerifyProc[EXP_MAXLEN_DS_EDI_VERIFY_PROC];

/* The Information Exchange charge information */
char Charge[EXP_MAXLEN_DS_CHARGE];

/* The type of acknowledgment */
char Ack[EXP_MAXLEN_DS_ACK];

/* The file retention period */
char Retention[EXP_MAXLEN_DS_RETENTION];

/* The delivery service type for the file */
char Priority[EXP_MAXLEN_DS_PRIORITY];

/* Reserved field */
char Reserved1[20];

/* The data type to send to Information Exchange */
char Datatype[EXP_MAXLEN_DS_DATATYPE];

/* The e-mail format option */
char Format[EXP_MAXLEN_DS_FORMAT];

/* The file description */
char Description[EXP_MAXLEN_DS_DESCRIPTION];

/* The data delimitation status */
char Delimited[EXP_MAXLEN_DS_DELIMITED];

/* The record format of the file to send */
char Recformat[EXP_MAXLEN_DS_RECFORMAT];

/* The record length of the file to send */
char Recordlen[EXP_MAXLEN_DS_RECORDLEN];

/* The prevent blanket-receive option*/
char Selectrcv[EXP_MAXLEN_DS_SELECTRCV];

/* The destination file name */
char Destfile[EXP_MAXLEN_DS_DESTFILE];
```c
/* The destination location */
char Destloc[EXP_MAXLEN_DS_DESTLOC];

/* The test or normal file indicator */
char Mode[EXP_MAXLEN_DS_MODE];

/* The message name */
char MsgName[EXP_MAXLEN_DS_MSG_NAME];

/* The message sequence number */
char MsgSeqNo[EXP_MAXLEN_DS_MSG_SEQ_NO];

/* The translation table name */
char XlateTbl[EXP_MAXLEN_DS_XLATE_TBL];

/* The option to send data as EDI or unformatted */
char Sendedi[EXP_MAXLEN_DS_SENDEDI];
};
```
Input field definitions for ExpDoSendOrdrReqStruct

For a list of the output fields, see “Output field definitions for ExpSendOrdrDoneRspStruct” on page 552.

**ProjName**
The project name. Because you can have several projects open at one time, the project name identifies which one to use for this request.

- **Max length:** EXP_MAXLEN_PJ_PROJ_NAME (20)
- **Min length:** EXP_MINLEN_PJ_PROJ_NAME (1)
- **Default value:** No
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, the project must be open.

**IeSysid**
The Information Exchange system where the account ID and user ID are registered. If you specify IeSysid, you must specify IeMboxId to create the Information Exchange address.

- **Max length:** EXP_MAXLEN_DS_IE_SYSID (3)
- **Min length:** EXP_MINLEN_DS_IE_SYSID (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**IeMboxId**
The Information Exchange account ID and user ID. Each value is specified with eight fixed characters and padded on the right with blanks, if needed. The first eight characters are the account ID; the next eight are the user ID.

- **Max length:** EXP_MAXLEN_DS_IE_MBOX_ID (16)
- **Min length:** EXP_MINLEN_DS_IE_MBOX_ID (9)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**TableType**
The alias table type for an Information Exchange alias table. If you specify TableType, you must specify AliasTbl and Aliasname.

- **Max length:** EXP_MAXLEN_DS_TABLE_TYPE (1)
- **Min length:** EXP_MINLEN_DS_TABLE_TYPE (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - G Alias table for global account.
  - O Alias table for a specific account.
  - P Alias table for private use.

**AliasTbl**
The alias table name for an Information Exchange alias table. If you specify AliasTbl, you must specify TableType and Aliasname.

- **Max length:** EXP_MAXLEN_DS_ALIAS_TBL (3)
- **Min length:** EXP_MINLEN_DS_ALIAS_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Aliasname**
This is an alias name in an Information Exchange alias table. If you specify Aliasname, you must specify TableType and AliasTbl.

- **Max length:** EXP_MAXLEN_DS_ALIASNAME (16)
- **Min length:** EXP_MINLEN_DS_ALIASNAME (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
Listname
The name of the list.

| Max length: | EXP_MAXLEN_DS_LISTNAME (8) |
| Min length: | EXP_MINLEN_DS_LISTNAME (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |

Nickname
This is a nickname from the address book. Use the Nickname with the AddrType to determine which network address to use to send or receive data. If you specify Nickname, you must specify AddrType.

| Max length: | EXP_MAXLEN_DS_NICKNAME (20) |
| Min length: | EXP_MINLEN_DS_NICKNAME (1) |
| Default value: | Blank |
| Required: | No |
| Case sensitive: | No |
| Validation: | No |

AddrType
The address type to use to send or receive data. Use the AddrType with the Nickname to determine which network address to use to send or receive data. If you specify AddrType, you must specify Nickname.

| Max length: | EXP_MAXLEN_DS_ADDR_TYPE (1) |
| Min length: | EXP_MINLEN_DS_ADDR_TYPE (1) |
| Default value: | E |
| Required: | No |
| Case sensitive: | Yes |
| Validation: | The valid values are: |
| A   Use the alias name and the alias table. |
| D   Use a distribution list with the name specified in the Nickname field. |
| E   Use the Information Exchange address. |
| X   Use the Mail Exchange address. |
**MsgClass**
The Information Exchange user message class specified by the sender.

- **Max length:** EXP_MAXLEN_DS_MSG_CLASS (8)
- **Min length:** EXP_MINLEN_DS_MSG_CLASS (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No

**Fileid**
The file name to send.

- **Max length:** EXP_MAXLEN_DS_FILEID (65)
- **Min length:** EXP_MINLEN_DS_FILEID (1)
- **Default value:** Blank
- **Required:** Yes
- **Case sensitive:** No
- **Validation:** Yes, as a file path.

**Verify**
This option indicates whether the receiver verification takes place before sending the file.

- **Max length:** EXP_MAXLEN_DS_VERIFY (1)
- **Min length:** EXP_MINLEN_DS_VERIFY (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F  Verify that the receiver exists. If Information Exchange cannot verify, send data anyway.
  - N  Do not verify the address.
  - Y  Verify the address. If the address is invalid or cannot be verified, do not send the data.
**EdiVerifyProc**

This option indicates what action to take if the address verification fails while sending envelopes from a file with multiple envelopes.

- **Max length:** EXP_MAXLEN_DS_EDI_VERIFY_PROC (1)
- **Min length:** EXP_MINLEN_DS_EDI_VERIFY_PROC (1)
- **Default value:** S
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **C** Continue processing.
  - **S** Stop processing this file and continue with next command.

**Charge**

This indicates to Information Exchange how the sender wants to pay the file charges.

- **Max length:** EXP_MAXLEN_DS_CHARGE (1)
- **Min length:** EXP_MINLEN_DS_CHARGE (1)
- **Default value:** 3
- **Required:** No
- **Case sensitive:** No
- **Validation:** The valid values are:
  1. The receiver pays all charges.
  2. The receiver pays all charges if agreed to by the receiver. Otherwise, the sender and the receiver split the charges.
  3. The receiver pays all charges if agreed to by the receiver. If not, the sender and receiver split the charges if agreed to by the receiver. Otherwise, the sender pays all charges. This is the default.
  4. The sender and the receiver split all charges if agreed to by the receiver. Otherwise, the sender pays all charges.
  5. The sender and receiver split all charges.
  6. The sender pays all charges.
**Ack**
This indicates the type of acknowledgment. Information Exchange puts these acknowledgments into the sender’s mailbox.

Max length: EXP_MAXLEN_DS_ACK (1)
Min length: EXP_MINLEN_DS_ACK (1)
Default value: N
Required: No
Case sensitive: Yes
Validation: The valid values are:

A  Information Exchange provides purge acknowledgments.
B  Information Exchange provides receipt and delivery acknowledgments.
C  Information Exchange provides receipt and purge acknowledgments.
D  Information Exchange provides delivery acknowledgments.
E  Information Exchange provides purge or delivery acknowledgments.
F  Information Exchange provides receipt and either purge or delivery acknowledgments.
N  Information Exchange does not provide acknowledgments.
R  Information Exchange provides receipt acknowledgments.
Chapter 5. Writing the interface for Information Exchange

Input field definitions for ExpDoSendOrdReqStruct

**Retention**
The number of days Information Exchange keeps the file in the mailbox if no one receives it. The default value of 0 indicates that Information Exchange should use the system default, which varies with the Information Exchange installation. Currently, the default in the U.S. is 30 days.

- **Max length:** EXP_MAXLEN_DS_RETENTION (3)
- **Min length:** EXP_MINLEN_DS_RETENTION (1)
- **Max value:** EXP_MAXVAL_DS_RETENTION (180)
- **Min value:** EXP_MINVAL_DS_RETENTION (0)
- **Default value:** 0
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.

**Priority**
This is the delivery-service class for the file to be sent.

- **Max length:** EXP_MAXLEN_DS_PRIORITY (1)
- **Min length:** EXP_MINLEN_DS_PRIORITY (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Normal priority. The file is received in the order placed in the mailbox. The normal priority is first-in, first-out.
  - **H** High priority. The file is placed at the beginning of the receiver’s queue.
  - **X** Express priority. Express delivery to users who have continuous receive capability and are currently in session with Information Exchange. Otherwise, the file is discarded.

**Reserved1**
This is a reserved field; leave it blank.
Length = 20.
**Datatype**
This indicates whether the data sent is text or binary. Binary data is not translated to EBCDIC and is sent as is.

- **Max length:** EXP_MAXLEN_DS_DATATYPE (1)
- **Min length:** EXP_MINLEN_DS_DATATYPE (1)
- **Default value:** T
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - B Binary data.
  - E Supported EDI data.
  - T Text data.

**Format**
This indicates whether Expedite for Windows should format the data as e-mail.

- **Max length:** EXP_MAXLEN_DS_FORMAT (1)
- **Min length:** EXP_MINLEN_DS_FORMAT (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Do not format as e-mail.
  - Y Format the text data as 79-byte records, padded on the right with blanks. The default, user class is FFMSG001.

**Description**
This is a free-format file description.

- **Max length:** EXP_MAXLEN_DS_DESCRIPTION (79)
- **Min length:** EXP_MINLEN_DS_DESCRIPTION (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** No
**Delimited**
This indicates whether or not the data is delimited with CRLF characters.

- **Max length:** EXP_MAXLEN_DS_DELIMITED (1)
- **Min length:** EXP_MINLEN_DS_DELIMITED (1)
- **Default value:** Y
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - N Data is not delimited by CRLF characters.
  - Y Data is delimited with CRLF characters.

**Recformat**
The file’s record format. This information is stored in the CDH and does not affect processing on any system, except VM.

- **Max length:** EXP_MAXLEN_DS_RECFORMAT (1)
- **Min length:** EXP_MINLEN_DS_RECFORMAT (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - F Fixed-record format.
  - N Format is unknown.
  - V Variable-record format.

**Recordlen**
The file’s record length. A value of 0 indicates the record length is unknown.

- **Max length:** EXP_MAXLEN_DS_RECORDLEN (5)
- **Min length:** EXP_MINLEN_DS_RECORDLEN (1)
- **Max value:** EXP_MAXVAL_DS_RECORDLEN (65535)
- **Min value:** EXP_MINVAL_DS_RECORDLEN (0)
- **Default value:** 0
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, for numeric data.
Selectrcv
This indicates how the receiver can retrieve the file.

Max length: EXP_MAXLEN_DS_SELECTRCV (1)
Min length: EXP_MINLEN_DS_SELECTRCV (1)
Default value: N
Required: No
Case sensitive: Yes
Validation: The valid values are:
N  The receiver can retrieve the file as usual.
Y  The receiver must use msgkey, account ID and user ID, or class to receive this file.

Destfile
Indicates the name to use in the common data header (CDH) as the original file name. By default, Expedite for Windows will use the file name specified in the send order for Destfile in the CDH. If the receiver specifies “Use original file name” (Origfile = Y) on the receive order, an Expedite product will attempt to use this file name, rather than the file name specified on the receive order, to store the data.

Max length: EXP_MAXLEN_DS_DESTFILE (54)
Min length: EXP_MINLEN_DS_DESTFILE (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No

Destloc
Indicates the file path to use in the CDH as the original file location. By default, Expedite for Windows will use the file path specified on the send order for Destloc in the CDH. Expedite products record this information for the receiver, but do not use the location to store the file. When storing received data, Expedite for Windows always uses the file path specified on the receive order.

Max length: EXP_MAXLEN_DS_DESTLOC (65)
Min length: EXP_MINLEN_DS_DESTLOC (1)
Default value: Blank
Required: No
Case sensitive: No
Validation: No
Chapter 5. Writing the interface for Information Exchange

Input field definitions for ExpDoSendOrdReqStruct

**Mode**
This is a sender-specified character indicating the file mode.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_DS_MODE (1)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_DS_MODE (1)</td>
</tr>
<tr>
<td>Default value</td>
<td>N</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>Yes</td>
</tr>
<tr>
<td>Validation</td>
<td>The valid values are:</td>
</tr>
<tr>
<td></td>
<td>N Normal mode.</td>
</tr>
<tr>
<td></td>
<td>T Test mode.</td>
</tr>
</tbody>
</table>

**MsgName**
The message name specified in the SEND command. If you sent EDI data and did not specify a message name, Expedite for Windows uses the value from the EDI data.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_DS_MSG_NAME (20)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_DS_MSG_NAME (1)</td>
</tr>
<tr>
<td>Default value</td>
<td>Blank</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
</tr>
</tbody>
</table>

**MsgSeqNo**
The user-specified message sequence number or a number that Expedite for Windows assigns for EDI data.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max length</td>
<td>EXP_MAXLEN_XS_MSG_SEQ_NO (5)</td>
</tr>
<tr>
<td>Min length</td>
<td>EXP_MINLEN_XS_MSG_SEQ_NO (1)</td>
</tr>
<tr>
<td>Default value</td>
<td>Blank</td>
</tr>
<tr>
<td>Required</td>
<td>No</td>
</tr>
<tr>
<td>Case sensitive</td>
<td>No</td>
</tr>
<tr>
<td>Validation</td>
<td>No</td>
</tr>
</tbody>
</table>
**XlateTbl**
The translation table name.

- **Max length:** EXP_MAXLEN_DS_XLATE_TBL (8)
- **Min length:** EXP_MINLEN_DS_XLATE_TBL (1)
- **Default value:** Blank
- **Required:** No
- **Case sensitive:** No
- **Validation:** Yes, it must contain the file name of the translation table without the XLT extension.

**Sendedi**
The option to send the data as EDI or unformatted.

- **Max length:** EXP_MAXLEN_DS_SENDEDI (1)
- **Min length:** EXP_MINLEN_DS_SENDEDI (1)
- **Default value:** N
- **Required:** No
- **Case sensitive:** Yes
- **Validation:** The valid values are:
  - **N** Send as unformatted data.
  - **Y** Send as EDI data.
Chapter 5. Writing the interface for Information Exchange

Output arguments for ExpDoSendOrdr

Output arguments for ExpDoSendOrdr

```
struct ExpSendOrdrDoneRspStruct
{
    /* The send receipt name */
    char RcptName[EXP_MAXLEN_XS_RCPT_NAME];

    /* The status of the processed order */
    char StatusCode[EXP_MAXLEN_XS_STATUS_CODE];

    /* The Expedite for Windows session-start date */
    char PstartDate[EXP_MAXLEN_XS_PSTART_DATE];

    /* The Expedite for Windows session end date */
    char PendDate[EXP_MAXLEN_XS_PEND_DATE];

    /* The number of files sent */
    char FilesSent[EXP_MAXLEN_XS_FILES_SENT];

};
```
Output field definitions for ExpSendOrdrDoneRspStruct

The output can include any or all of these fields.

**RcptName**
The send receipt name.

- Max length: EXP_MAXLEN_XS_RCPT_NAME (20)
- Min length: EXP_MINLEN_XS_RCPT_NAME (1)
- Contents: Printable data.

**StatusCode**
The status results of a processed order.

- Max length: EXP_MAXLEN_XS_STATUS_CODE (5)
- Min length: EXP_MINLEN_XS_STATUS_CODE (1)
- Write access: Read only
- Contents: Numeric data.

**PstartDate**
The processing start date and time.

- Max length: EXP_MAXLEN_XS_PSTART_DATE (14)
- Min length: EXP_MINLEN_XS_PSTART_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**PendDate**
The processing end date and time.

- Max length: EXP_MAXLEN_XS_PEND_DATE (14)
- Min length: EXP_MINLEN_XS_PEND_DATE (14)
- Contents: The date and time in format YYYYMMDDHHMMSS.

**FilesSent**
The number of files sent as a result of processing this order.

- Max length: EXP_MAXLEN_XS_FILES_SENT (5)
- Min length: EXP_MINLEN_XS_FILES_SENT (1)
- Contents: Numeric data.
The Java interface classes

The Expedite for Windows Java interface enables Java programmers to write Java programs to control Expedite for Windows through the existing C-language interface.

This appendix contains the database classes and methods for each class for the Java interface. For information on programming to the Java interface, see the *Software Development Kit Programming Guide*.

Database classes, members, and methods

The database classes discussed in this chapter are listed below. Descriptions of these classes and lists of their related database members and methods, including the get and set methods, are in each section.

For the maximum and minimum lengths of fields, see the appropriate chapter for the corresponding C-language functions elsewhere in this book.

```
public class ExpAddress  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 555
public class ExpAddressList extends LinkedList . . . . . . . . . . . . . . . . . . . . . . . . 557
public class ExpCommProfile  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 558
public class ExpCommProfileList extends LinkedList  . . . . . . . . . . . . . . . . . . . 559
public class ExpDBLister extends LinkedList  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 560
public class ExpDistributionList  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 563
public class ExpDistListList extends LinkedList  . . . . . . . . . . . . . . . . . . . . . . . . 564
public class ExpDropBox  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 565
public class ExpDropBoxList extends LinkedList  . . . . . . . . . . . . . . . . . . . . . . . . 567
public class ExpException extends Exception  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 568
public class ExpIeLogon  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 569
public class ExpIeLogonList extends LinkedList  . . . . . . . . . . . . . . . . . . . . . . . . 570
```
Software Development Kit Programming Reference

public class ExpJni ................................................................. 571
public class ExpMailboxItem .................................................. 572
public class ExpMailboxItemList extends LinkedList ................... 573
public class ExpOrderList extends LinkedList ............................. 574
public class ExpProject ......................................................... 575
public class ExpProjectList extends LinkedList ........................... 576
public class ExpReceiveOrder .................................................. 577
public class ExpReceiveReceipt ............................................... 579
public class ExpReceiptList extends LinkedList ........................... 581
public class ExpReceiveOrderList extends LinkedList ................. 582
public class ExpReceiveReceiptList extends LinkedList ............... 583
public class ExpSendOrder ...................................................... 584
public class ExpSendReceipt .................................................. 586
public class ExpSendOrderList extends LinkedList ..................... 587
public class ExpSendReceiptList extends LinkedList ................... 588
public class ExpSessionReceipt ............................................... 589
public class ExpSessionReceiptList extends LinkedList ............... 590
public class ExpSession ......................................................... 591
public class ExpTradingProfile .............................................. 593
public class ExpTradingProfileList extends LinkedList ............... 594
public class ExpAddress

The ExpAddress class is used to interact with the address records in the Expedite for Windows database.

Data members

private String nickname
private String firstname
private String lastname
private String title
private String position
private String orgName
private String city
private String street
private String state
private String country
private String zip
private String ieMboxId
private String ieSysid
private String imxIea
private String tableType
private String aliasTbl
private String aliasname
private String voicePhone1
private String voicePhone2
private String faxNum

Methods

“get” and “set” methods for all data members
public int add() throws ExpException
public int delete() throws ExpException
public static int delete(String theNickname) throws ExpException
public int replace() throws ExpException
public int assignToDistributionList(String theDistListName) throws ExpException
public static int assignToDistributionList(String theDistListName, String theNickname) throws ExpException
public int unassignFromDistributionList(String theDistListName) throws ExpException
public static int unassignFromDistributionList(String theDistListName,
String theNickname) throws ExpException
public int assignToTradingProfile(String theTradProfName)
    throws ExpException
public static int assignToTradingProfile(String theTradProfName,
    String theNickname) throws ExpException
public int unassignFromTradingProfile(String theTradProfName)
    throws ExpException
public static int unassignFromTradingProfile(String theTradProfName,
    String theNickname) throws ExpException
public class ExpAddressList extends LinkedList

The ExpAddressList class is used when working with multiple address records in the Expedite for Windows database.

Data members

none

Methods

public ExpAddress findByNickname(String theNickName)
public class ExpCommProfile

The ExpCommProfile class is used to interact with the communications profile records in the Expedite for Windows database.

Data members

private String cprofName
private String commtype
private String reconnects
private String dialCount
private String dialProfile

Methods

“get” and “set” methods for all data members
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theCprofName) throws ExpException
public class ExpCommProfileList extends LinkedList
   The ExpCommProfileList class is used when working with multiple records in the Expedite for Windows database.

Data members
   none

Methods
   public ExpCommProfile findByCprofName(String theCprofName)
**public class ExpDBLister extends LinkedList**

The ExpDBLister class is used to obtain lists from the Expedite for Windows database. Each list method has two variations. The method that does not require any parameters returns all the requested objects from the Expedite for Windows database in the default order. The second variation of the methods takes a list request structure as a parameter. By setting the data members of the list request structure, selective lists and sorted lists can be obtained from the database. Three methods have been implemented to obtain selected objects from the database: `getAddressByNickname`, `getSendReceiptByRcptName`, and `getReceiveReceiptByRcptName`. Use these methods to obtain address and receipt objects directly from the database by specifying the object's name.

**Data members**

none

**Methods**

```java
public static ExpAddressList getAddresses() throws ExpException
public static ExpAddressList getAddresses(ExpListReqStruct theRequest) throws ExpException
public static ExpAddress getAddressByNickname(String theNickname) throws ExpException
public static ExpAddressList getAddressesOnDistList(String theDistListName) throws ExpException
public static ExpAddressList getAddressesOnDistList(String theDistListName, ExpListItemsReqStruct theRequest) throws ExpException
public static ExpAddressList getAddressesOnTradingProfile(String theTradProfName) throws ExpException
public static ExpAddressList getAddressesOnTradingProfile(String theTradProfName, ExpListItemsReqStruct theRequest) throws ExpException
public static ExpCommProfileList getCommProfiles() throws ExpException
public static ExpCommProfileList getCommProfiles(ExpListReqStruct theRequest) throws ExpException
public static ExpDistListList getDistributionLists() throws ExpException
public static ExpDistListList getDistributionLists(ExpListReqStruct theRequest) throws ExpException
```
public static ExpDistListList getDistListsOnTradingProfile(String theTradProfName) throws ExpException
public static ExpDistListList getDistListsOnTradingProfile(String theTradProfName, ExpListItemsReqStruct theRequest) throws ExpException
public static ExpDropBoxList getDropBoxes() throws ExpException
public static ExpDropBoxList getDropBoxes(ExpListReqStruct theRequest) throws ExpException
public static ExpIeLogonList getIeLogons() throws ExpException
public static ExpIeLogonList getIeLogons(ExpListReqStruct theRequest) throws ExpException
public static ExpOrderList getOrders() throws ExpException
public static ExpOrderList getOrders(ExpListReqStruct theRequest) throws ExpException
public static ExpOrderList getOrdersOnDropBox(String theDropBoxName) throws ExpException
public static ExpOrderList getOrdersOnDropBox(String theDropBoxName, ExpListItemsReqStruct theRequest) throws ExpException
public static ExpProjectList getProjects() throws ExpException
public static ExpProjectList getProjects(ExpListProjReqStruct theRequest) throws ExpException
public static ExpProjectList getProjectsOnCommProfile(String theCommProfName) throws ExpException
public static ExpProjectList getProjectsOnCommProfile(String theCommProfName, ExpListProjReqStruct theRequest) throws ExpException
public static ExpReceiptList getReceiptsOnSessionReceipt(String theSessionReceiptName) throws ExpException
public static ExpReceiptList getReceiptsOnSessionReceipt(String theSessionReceiptName, ExpListItemsReqStruct theRequest) throws ExpException
public static ExpReceiptList getReceiptsOnReceiptShelf() throws ExpException
public static ExpReceiptList getReceiptsOnReceiptShelf(ExpListItemsReqStruct theRequest) throws ExpException
public static ExpReceiveOrderList getReceiveOrders() throws ExpException
public static ExpReceiveOrderList getReceiveOrders(ExpListReqStruct theRequest) throws ExpException
public static ExpReceiveReceiptList getReceiveReceipts() throws ExpException
public static ExpReceiveReceiptList getReceiveReceipts(ExpListReqStruct theRequest) throws ExpException
public static ExpReceiveReceipt getReceiveReceiptByRcptName(String theRcptName) throws ExpException
public static ExpSendOrderList getSendOrders() throws ExpException
public static ExpSendOrderList getSendOrders(ExpListReqStruct theRequest) throws ExpException
public static ExpSendReceiptList getSendReceipts() throws ExpException
public static ExpSendReceiptList getSendReceipts(ExpListReqStruct theRequest) throws ExpException
public static ExpSendReceipt getSendReceiptByRcptName(String theRcptName) throws ExpException
public static ExpSessionReceiptList getSessionReceipts() throws ExpException
public static ExpSessionReceiptList getSessionReceipts(ExpListReqStruct theRequest) throws ExpException
public static ExpTradingProfileList getTradingProfiles() throws ExpException
public static ExpTradingProfileList getTradingProfiles(ExpListReqStruct theRequest) throws ExpException
public class ExpDistributionList

The ExpDistributionList class is used to interact with the distribution list records in the Expedite for Windows database.

Data members

public String distListName
public String description

Methods

“get” and “set” methods for all data members
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theDistListName) throws ExpException
public int assignToTradingProfile(String theTradProfName)
    throws ExpException
public int assignToTradingProfile(String theTradProfName,
    String theDistListName) throws ExpException
public int unassignFromTradingProfile(String theTradProfName)
    throws ExpException
public static int unassignFromTradingProfile(String theTradProfName,
    String theDistListName) throws ExpException
public class ExpDistListList extends LinkedList

The ExpDistListList class is used when working with multiple records in the Expedite for Windows database.

Data members
none

Methods

public ExpDistributionList findByDistListName(String theDistListName)
public class ExpDropBox

The ExpDropBox class is used to interact with the drop box records in the Expedite for Windows database.

Data members

private String dropBoxName
private String ieMboxIdName
private String emailOpt
private String sysmsgOpt
private String ordersOpt
private String appStatus
private String dropRecovery
private String sessRecovery
private String commitdata
private String maxmsgs
private String msgsize
private String xlateTbl
private String overwrite
private String timeZone

Because the following data members are set when add or replace is called, there are no set methods for these data members.

“get” and “set” methods for all data members
private String status*
private String ieMboxId*
private String numAssigned*

Because the following data members are set when process is called, there are no set methods for these data members.

“get” and “set” methods for all data members
private static String srcptName*
private static String startDate*
private static String endDate*
private static String sesskey*
private static String ierevision*
private static String ierelease*
private static String lastSess*
private static String ierespcode*
private static String statusCode*
Methods

“get” and “set” methods for all data members

public int process(String theProjName) throws ExpException
public static int process(String theProjName, String theDropBoxName) throws ExpException

public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theDropBoxName) throws ExpException
public class ExpDropBoxList extends LinkedList

The ExpDropBoxList class is used when working with multiple records in the Expedite for Windows database.

Data members

none

Methods

public ExpDropBox findByDropBoxName(String theDropBoxName)
public class ExpException extends Exception

The ExpException class is used when exceptions are thrown by the JNI DLL.

Data members

private ErrorTextRspStruct itsError;

Methods

“get” and “set” methods for all data members
public String getReasonCode()
public String getErrorMsg()
public String getErrorTxt()
public String getResponse()
public class ExpIeLogon

The ExpIeLogon class is used to interact with the Information Exchange Logon records in the Expedite for Windows database.

Data members

private String name
private String account
private String password
private String status
private String description
private String numAssigned
private String npassword

Methods

“get” and “set” methods for all data members
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theName) throws ExpException
public class ExpIeLogonList extends LinkedList

The ExpIeLogonList class is used when working with multiple records in the Expedite for Windows database.

Data members
none

Methods
public ExpIeLogon findByName(String theName)
public class ExpJni

The ExpJni class is the interface class to the Java native methods. These methods are not called directly by end user programs except to open and close the C-language interface. The class has one data member that stores the current opened project.

Data members

private static String currentProject

Methods

protected static String getCurrentProject
public static native int jniOpenCInterface() throws ExpException
public static native int jniCloseCInterface() throws ExpException
public class ExpMailboxItem

The ExpMailboxItem class is used to store the data returned from the ExpSession.doQuery method call.

Data members

private String statusCode
private String ieSysid
private String ieMboxId
private String msgkey
private String msgClass
private String msgName
private String msgSeqNo
private String uniqueid
private String msgDate
private String length
private String datatype
private String editype
private String sysname
private String syslevel
private String senderfile
private String senderloc
private String description
private String filedate
private String recfm
private String reclen
private String recdlm
private String systype
private String sysver
private String comsw
private String comver
private String comfile
private String mode

Methods

“get” and “set” methods for all data members
Because the data members are set when doQuery is called, this class only contains “get” methods for each data member.
public class ExpMailboxItemList extends LinkedList

The ExpMailboxItemList class is a linked list used to store the ExpMailboxItem objects when the ExpSession.doQuery method is called.

Data members

none

Methods

none
public class ExpOrderList extends LinkedList
    
The ExpOrderList class is used when working with multiple records in the Expedite for Windows database.

Data members
    none

Methods
    public Object findByOrdrName(String theOrdrName)
public class ExpProject

The ExpProject class is used to interact with the project records in the Expedite for Windows database.

Data members

private String projName
private String creator
private String description
private String dateModified
private String readPswd
private String writePswd
private String rcpRetain
private String location
private String status
private String path

private String pswdNeeded*
private static String accessLevel*

Methods

“get” and “set” methods for all data members
*pswdNeeded only has a get method. It is set by listing the project.
*accessLevel only has a get method. It is set when the project is opened.

public static int open(String theProjName, String theProjPswd) throws ExpException

public static int close() throws ExpException

public int add() throws ExpException

public int delete() throws ExpException

public static int delete(String theProjectName) throws ExpException

public int replace() throws ExpException

public int assignToCommProfile(String theCommProfileName) throws ExpException

public static int assignToCommProfile(String theProjectName, String theCommProfileName) throws ExpException

public int unassignFromCommProfile(String theCommProfileName) throws ExpException

public static int unassignFromCommProfile(String theProjectName, String theCommProfileName) throws ExpException
public class ExpProjectList extends LinkedList

The ExpProjectList class is used when working with multiple records in the Expedite for Windows database.

Data members

none

Methods

public ExpProject findByProjName(String theProjName)
public class ExpReceiveOrder

The ExpReceiveOrder class is used to interact with the receive order records in the Expedite for Windows database.

Data members

private String ordrName
private String ordrDescr
private String ieSysid
private String ieMboxId
private String tableType
private String aliasTbl
private String aliasname
private String listname
private String nickname
private String addrType
private String msgClass
private String fileid
private String archiveid
private String allfiles
private String multfiles
private String origfile
private String format
private String recordsize
private String processlen
private String autoedi
private String ediopt
private String requeued
private String removeeof
private String nonedioonly
private String edionly
private String overwrite
private String waittime
private String msgkey
private String startDate
private String endDate
private String glTimeZone
private String xlateTbl
private String receiveedi
private String rcptName*
private String statusCode*
private String pstartDate*
private String pendDate*
private String filesRcvd*

Methods

“get” and “set” methods for all data members
  *rcptName, statusCode, pstartDate, pendDate, and filesRcvd only have
get methods. They are set by the doOrder method.
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theOrderName) throws ExpException
public int assignToDropBox(String theDropBoxName) throws ExpException
public static int assignToDropBox(String theReceiveOrderName,
                              String theDropBoxName) throws ExpException
public int unassignFromDropBox(String theDropBoxName) throws ExpException
public static int unassignFromDropBox(String theReceiveOrderName,
                              String theDropBoxName) throws ExpException
public int doOrder() throws ExpException
public class ExpReceiveReceipt

The ExpReceiveReceipt class is used to interact with the receive receipt records in the Expedite for Windows database.

Data members

private String rcptName
private String statusCode
private String pstartDate
private String pendDate
private String ordrName
private String ieSysid
private String ieMboxId
private String tableType
private String aliasTbl
private String aliasname
private String msgClass
private String fileid
private String uniqueid
private String msgDate
private String length
private String charge
private String ack
private String priority
private String datatype
private String receiver
private String recvqual
private String sender
private String sendqual
private String controlnum
private String editype
private String sysname
private String syslevel
private String senderfile
private String senderloc
private String description
private String filedate
private String ieMsgSeqNo
private String sessionkey
private String archiveid
private String recfm
private String reclen
private String recdlm
private String delimited
private String systype
private String sysver
private String comsw
private String comver
private String comfile
private String codepage
private String dcmprc
private String mode
private String msgName
private String msgSeqNo
private String xlateTbl
private String cmmtCode
private String sequenceNum
private String ordrStatus
private String listname
private String nickname
private String addrType

**Methods**

“get” methods for each data member.
The “set” methods are not required because the data members of the receipts are set by listing them.
public class ExpReceiptList extends LinkedList

The ExpReceiptList class is used when working with multiple records in the Expedite for Windows database.

Data members

none

Methods

none
public class ExpReceiveOrderList extends LinkedList
    The ExpReceiveOrderList class is used when working with multiple records in the Expedite for Windows database.

Data members
    none

Methods
    public ExpReceiveOrder findByOrdrName(String theOrdrName)
public class ExpReceiveReceiptList extends LinkedList

The ExpReceiveReceiptList class is used when working with multiple records in the Expedite for Windows database.

Data members
none

Methods
public ExpReceiveReceipt findByRcptName(String theRcptName)
public class ExpSendOrder

The ExpSendOrder class is used to interact with the send order records in the Expedite for Windows database.

Data members

private String ordrName
private String ordrDescrip
private String ieSysid
private String ieMboxId
private String tableType
private String aliasTbl
private String aliasname
private String listname
private String nickname
private String addrType
private String msgClass
private String fileid
private String verify
private String ediVerifyProc
private String charge
private String ack
private String retention
private String priority
private String datatype
private String format
private String description
private String delimited
private String recformat
private String recordlen
private String selectrcv
private String destfile
private String destloc
private String mode
private String msgName
private String msgSeqNo
private String xlateTbl
private String sendedi

private String rcptName*
private String statusCode*
private String pstartDate*
private String pendDate*
private String filesSent*

Methods

“get” and “set” methods for all data members
*rcptName, statusCode, pstartDate, pendDate, and filesSent only have get methods. They are set by the doOrder method.
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theOrderName) throws ExpException
public int assignToDropBox(String theDropBoxName) throws ExpException
public static int assignToDropBox(String theSendOrderName, String theDropBoxName) throws ExpException
public int unassignFromDropBox(String theDropBoxName) throws ExpException
public static int unassignFromDropBox(String theSendOrderName, String theDropBoxName) throws ExpException
public int doOrder() throws ExpException
public class ExpSendReceipt

The ExpSendReceipt class is used to interact with the send receipt records in the Expedite for Windows database.

Data members

private String rcptName
private String statusCode
private String pstartDate
private String pendDate
private String ordrName
private String ieSysid
private String ieMboxId
private String tableType
private String aliasTbl
private String aliasname
private String listname
private String msgClass
private String uniqueid
private String length
private String editype
private String destination
private String qualifier
private String controlnum
private String msgName
private String msgSeqNo
private String sentCode
private String cmmtCode
private String sequenceNum
private String fileid
private String ordrStatus
private String nickname
private String addrType

Methods

“get” methods for each data member. The “set” methods are not required because the data members of the receipts are set by listing them.
public class ExpSendOrderList extends LinkedList

The ExpSendOrderList class is used when working with multiple records in the Expedite for Windows database.

Data members
none

Methods
public ExpSendOrder findByOrdrName(String theOrdrName)
public class ExpSendReceiptList extends LinkedList

    The ExpSendReceiptList class is used when working with multiple records in the Expedite for Windows database.

Data members

    none

Methods

    public ExpSendReceipt findByRcptName(String theRcptName)
public class ExpSessionReceipt

The ExpSessionReceipt class is used to interact with the session receipt records in the Expedite for Windows database.

Data members

private String srcptName
private String statusCode
private String dropBoxName
private String startDate
private String endDate
private String sesskey
private String ieversion
private String ierelease
private String lastSess
private String ierespcode
private String startStatus
private String endStatus
private String startOrdrStatus
private String endOrdrStatus

Methods

“get” methods for each data member. The “set” methods are not required because the data members of the receipts are set by listing them.

custom int delete() throws ExpException
public static int delete(String theSrcptName) throws ExpException
public static int recover(String theSrcptName, String theRecovery,
                          ExpSessRcptRecoveredRspStruct theResponse)
                          throws ExpException
public int recover(String theRecovery,
                   ExpSessRcptRecoveredRspStruct theResponse)
                   throws ExpException
public class ExpSessionReceiptList extends LinkedList

The ExpSessionReceiptList class is used when working with multiple records in the Expedite for Windows database.

Data members
none

Methods

public ExpSessionReceipt findBySrcptName(String theSrcptName)
public class ExpSession

The ExpSession class is used when starting and stopping sessions with the interactive commands. The methods for the interactive commands are ExpSession.doQuery, ExpSession.doPurge, ExpSendOrder.doOrder, and ExpReceiveOrder.doOrder. The start method must be called before any of the above methods are called and the end method must be called upon conclusion of the method calls.

Data members

private String ieMboxIdName
private String appStatus
private String sessRecovery
private String commitdata
private String maxmsgs
private String msgsize
private String xlateTbl
private String overwrite
private String check
private String timeZone
private String sesskey*
private String ieveresion*
private String ierelease*
private String lastSess*
private String ierespcode*
private String srctxtName*
private static String statusCode*
private String startStatus*
private String endStatus*
private String startOrdrStatus*
private String endOrdrStatus*

private String connectStat
private String sessStat
private String dropBoxName*
private String startDate*
private String endDate*
Methods

“get” and “set” methods for all data members
*The marked data members only contain get methods because their values are set by calling one or more of the object's methods.

public int start() throws ExpException
public int end() throws ExpException
public static ExpMailboxItemList doQuery() throws ExpException
public static int doPurge(String theMsgkey) throws ExpException
public class ExpTradingProfile

The ExpTradingProfile class is used to interact with the trading profile records in the Expedite for Windows database.

Data members

private String tprofName
private String charge
private String msgClass
private String msgName
private String msgSeqNo
private String ack
private String priority
private String retention
private String compressName

Methods

“get” and “set” methods for all data members
public int add() throws ExpException
public int replace() throws ExpException
public int delete() throws ExpException
public static int delete(String theTprofName) throws ExpException
public class ExpTradingProfileList extends LinkedList

The ExpTradingProfileList class is used when working with multiple records in the Expedite for Windows database.

Data members

none

Methods

public ExpTradingProfile findByTprofName(String theTprofName)
This glossary defines words as they are used in this book. If you are looking for a term and cannot find it here, see the *Dictionary of Computing* for additional definitions.

A

account. A unique identifier assigned to a group of users who work for the same company.

account ID. A name that identifies an account to a program, device, or system.

account profile. Data that describes the characteristics of a user or group of users.

acknowledgment. A response from Information Exchange that confirms whether files were delivered, received, purged, or any combination of these actions.

address. (1) A unique code assigned to a user connected to a network. (2) The location in the storage of a computer where data is stored.

address book. A repository for trading-partner information. The address book can contain nicknames (for Information Exchange addresses) and trading profiles.

alphanumeric. Pertaining to data that consists of letters, digits, and usually other characters, such as punctuation marks.

alias name. An alternate name used in place of an account and user ID. This is used with an alias table and alias table type.

alias table. A permanent file list of alternate names that resides in Information Exchange.

alias table type. A single character access level indicator used with an alias table name to identify an alias table. Values are G for global access, P for owner access, and O for organizational (account) access.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

API. See application programming interface.
application link. A set of files containing information that Request Manager uses to interact with different applications.

application programming interface. A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program.

archive. A place to store messages on a database for future reference.


attribute. A property or characteristic of one or more entities; for example, length, value, color, or intensity.

audit trail. Data, in the form of a logical path, linking a sequence of events used to trace and verify the transactions that have affected the contents of a record.

authorization level. The ability to do certain restricted functions.

application link. A set of files containing information that Request Manager uses to interact with different applications.

B

binary. Machine instructions that a person cannot read or enter from a computer keyboard.

C
carriage-return and line-feed characters (CRLF). A word processing formatting control that moves the printing or display point to the first position of the next line.

CDH. Common data header.

centralized alias table. Permanent tables that reside in Information Exchange and contain a centralized list of addresses. You can put a list of your trading partners’ addresses in this table rather than maintain destination tables in multiple locations.

A centralized alias table enables Information Exchange to resolve destinations because it contains a list of EDI destinations paired with Information Exchange destinations. Expedite for Windows searches this table for an EDI destination and then uses the corresponding Information Exchange destination as the actual address.

character. A letter, digit, or other symbol used as part of the organization, control, or representation of data.

checkpoint-level recovery. A method of restart and recovery within Expedite for Windows. A point where information about the status of a data transmission can be recovered so it can be restarted later.

class. In object-oriented design or programming, a model or template that can be instantiated to create objects with a common definition and, therefore, common properties, operations, and behavior. An object is an instance of a class.

command. A request from a terminal for the performance of an operation or the execution of a particular program.
command line. On a display screen, a display line on which only instructions to the operating system can be entered.

commit. The point in a session at which Expedite and Information Exchange record checkpoint information, such as number of characters and files transmitted so far, and other information needed to recover a session. A commit can occur during or after transmitting a file, depending on the checkpoint-level selected. Once a file is completely committed, Information Exchange will deliver the file to the recipient if you are sending, or it will purge the file from your mailbox if you are receiving. If a session fails, all uncommitted files are discarded, and the session can be resumed as the last commit checkpoint.

common data header (CDH). A set of control information about a file, which is sent to Information Exchange by some sending interfaces. When the file is received by the trading partner, the receiving interface can use the information in the CDH.

CRLF. Carriage-return and line-feed characters.

distribution list. A list of the addresses of users with whom a certain user communicates. It is used to send messages to several people without having to type their addresses.

dropoff box. The dropoff box represents an Information Exchange session. The dropoff box definition contains information about how Expedite should conduct the session with Information Exchange, such as recovery level, account and user ID for the mailbox, and your time zone. Once orders are created, they are assigned to a dropoff box for processing. The dropoff box is similar to the input file and the TRANSMIT and SESSION profile commands in Expedite Base.

command line. On a display screen, a display line on which only instructions to the operating system can be entered.

distribution list. A list of the addresses of users with whom a certain user communicates. It is used to send messages to several people without having to type their addresses.

dropoff box. The dropoff box represents an Information Exchange session. The dropoff box definition contains information about how Expedite should conduct the session with Information Exchange, such as recovery level, account and user ID for the mailbox, and your time zone. Once orders are created, they are assigned to a dropoff box for processing. The dropoff box is similar to the input file and the TRANSMIT and SESSION profile commands in Expedite Base.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

EDI. Electronic data interchange.

electronic data interchange (EDI). The process of sending specially formatted business documents directly from one computer to another electronically.

electronic mail (e-mail). Free formatted messages and formatted file correspondence sent from one computer to another.

EOF (end of file). A coded character recorded in a file to indicated the end of the file.

express-priority messages. Messages that are delivered immediately after they are received by Information Exchange; the recipient must be receiving messages to receive an express-priority message. If the receiver is not receiving messages, the file is discarded.
extended security option (ESO). An option that extended security users can specify in their profiles for stricter password security.

extended security users. Users with stricter security requirements, such as levels of password protection.

F

field. An area of a panel reserved for data of a certain type or length.

file. A named set of records stored or processed as a unit.

file-level recovery. A method of restart and recovery within Expedite for Windows; checkpoints are taken for each file sent and received.

I

Information Exchange. A communication service that allows users to send and receive information electronically.

Information Exchange Administration Services. An online, panel-driven product that the Information Exchange Service Administrator uses to perform administrative tasks for Information Exchange.

Information Exchange Service Administrator. The person who coordinates the use of Information Exchange in a company.

J

Java. An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

Java Development Kit (JDK). A software package that can be used to write, compile, debug, and run Java applets and applications.

L

library. A place to store information for an extended period of time. A library consists of a collection of files called library members.

library member. A named collection of records or statements in a library.
mailbox. (1) A database that contains records that represent orders and receipts for processed orders.

(1) Any piece of data that users send or receive. (2) The smallest subdivision of information that can be sent from one user to another. (3) An instruction or explanation on the screen that describes what the system is doing or warns that the system has detected an error.

message acknowledgment. A response from Information Exchange that lets users know whether their messages were delivered, received, purged, or various combinations of the three.

message class. A category, agreed upon by trading partners, that is used to group mail.

message group. A collection of messages that is treated as a single entity. A file is an example of a message group.

message header. The leading part of a message that contains information, such as the source or destination code of the message.

method. In object-oriented design or programming, the software that implements the behavior specified by an operation.

non EDI data. Rules defined by NIST to enable X.400 users to exchange binary files through the 1984 X.400 InterPersonal Messaging Services.

object. In object-oriented design or programming, a concrete realization of a class that consists of data and the operations associated with that data.

order. An Information Exchange request. The order object represents the Expedite Base commands in Expedite for Windows.

order shelf. When you create an order, Expedite stores the order as part of a set in the mailbox database, known as the order shelf.

order receipt. Expedite for Windows provides a confirmation of the results of order processing that corresponds to a record in the output file of Expedite.

organizational alias table. An alias table setup within an account.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

password. A combination of confidential characters that users enter when they log on, to prevent unauthorized access of their systems and data.

National Institute of Standards and Technology (NIST). In the United States, this was formerly the National Bureau of Standards.
permanent distribution list. A distribution list stored permanently in Information Exchange.

private alias. An alias that can be used only by the user who created it.

private alias table. An alias table setup for an individual user.

projects. A naming mechanism that allows multiple users or applications (one at a time) access to the same copy of Expedite for Windows.

receipt acknowledgment. A confirmation generated by Information Exchange when a file reaches the receiver’s mailbox after a successful session.

receipt shelf. When Expedite creates a receipt for order processing, it stores the receipt as part of a set in the mailbox database, known as the receipt shelf.

receiver. The user or users to whose mailboxes you are sending, or retrieving information.

receive-side charges. The charges that users incur when they receive messages through Information Exchange.

Request Manager. The component of Expedite for Windows that manages the databases and Information Exchange sessions.

reset. (1) A type of session recovery that indicates Expedite should mark unprocessed orders as pending and enable the mailbox ID. When the session is resumed, Expedite starts processing at the beginning of the first order marked pending. Note that if you were sending multiple EDI envelopes from a single file, and the session is interrupted, if you reset the session, it resumes sending envelopes from the beginning of the file. (2) A session recovery state that indicates the error that interrupted the session prevents the session from being resumed.

restart. (1) A session recovery state that indicates Expedite has recorded commit information during the session and can resume processing at the last checkpoint when the session is resumed. (2) To resume a session at the last checkpoint.

send-side charges. The charges that users incur when they send messages through Information Exchange.

session. The period of time during which a user of a terminal can communicate with an interactive system, usually, elapsed time between logon and logoff.

session-level recovery. A method of restart and recovery within Expedite for Windows; no files are committed until the session ends normally.

session receipt. An acknowledgment provided by Information Exchange that corresponds to session start and session end records in Expedite. The session receipt includes a set of order receipts, which provide information about the processing of a single order.

synchronous. A process that is completed within a regular or predictable time frame.
T

temporary distribution list. A distribution list that lasts only for the duration of an Information Exchange session.

trading partner. The business associates with whom users exchange information electronically.

trading partner list. A list of business associates that users can send information to and receive information from using Information Exchange.

trading partner profile. A list that defines which trading partner pays to send or receive messages as part of a set of default information for a trading partner to use to configure send orders.

user profile. A user description that includes account ID, user ID, and password information. The characteristics that designate how a user works with Information Exchange.

W

wildcard. A special character, such as a question mark, that can be used to represent one or more characters for pattern matching.

X


U

UCS. Uniform Communication Standard.

Uniform Communication Standard (UCS). A standard EDI format used in the grocery industry.

United Nations/Trade Data Interchange (UN/TDI). An EDI standard for administration, commerce, and transportation fields developed by the United Nations Economic Commission for Europe.

UN/TDI. United Nations/Trade Data Interchange.

user ID. A name that identifies a user to Information Exchange within an account.

user message class. A category used to group mail. This category is agreed upon by trading partners.
Index

A
adding
  an address 91
  distribution lists 157
  dropoff boxes 225
  logon IDs 63
  receive orders 274
  send orders 350
  trading profile 178
address book
  adding an address 91
  deleting an address 109
  distribution lists 157
function
  ExpAddAddr 91
  ExpAddDistList 157
  ExpAddTradProf 178
  ExpAsgAddrToDistList 105
  ExpAsgAddrToTradProf 107
  ExpAsgDistListToTradProf 160
  ExpCloseAdbk 90
  ExpDeleteAddr 109
  ExpDeleteDistList 162
  ExpDeleteTradProf 189
  ExpListAddr 111
  ExpListAddrsOnDistList 121
  ExpListAddrsOnTradProf 130
  ExpListDistList 164
  ExpListDistListsOnTradProf 169
  ExpListTradProf 191
  ExpOpenAdbk 89
  ExpReplaceAddr 139
  ExpReplaceDistList 173
  ExpReplaceTradProf 200
  ExpUsgAddrFromDistList 153
  ExpUsgAddrFromTradProf 155
  ExpUsgDistListFromTradProf 176
opening 89
  trading profiles 178
  writing the interface for 87
addresses 91
  deleting from a distribution list 109
  replacing information 139
  trading profile 109
  unassigning from trading profile 155
assigning
  address to distribution lists 105
  address to trading profile 107
  receive orders 302, 348
  send orders 378
assigning an address to distribution lists 160

B
batch mode, processing dropoff box 476
C

closing
   a mailbox 214
   projects 16
communications profiles 45
   assigning a project to 24
   replacing information 58
   unassigning from a project 43

D
database
   address book 87
   Information Exchange 473
   mailbox 211
   projects 11
deleting
   addresses from distribution list 162
   an address 109
   communications profiles 50
   distribution lists 162
   dropoff boxes 242
   logon IDs 68
   send orders 380
   session receipts 428
   trading profiles 189
distribution lists
   adding 157
   assigning an address to 105, 160
   deleting 162
   deleting an address 109
   replacing 173
   unassigning an address 153
   unassigning from trading profile 176
dropoff boxes 225
   adding 225
   deleting 242
   interactive processing 487
   processing 476
   receive orders 242
   replacing 257
   send orders 242

E

ending interactive processing 481
error information 4
Expedite
   RECEIVE record 518
   SENT record 536

F

function name example xi
function prototype
   ExpAddAddr 91
   ExpAddCommProf 45
   ExpAddDistList 157
   ExpAddDropBox 225
   ExpAddField 6
   ExpAddIeLogon 63
   ExpAddProj 17
   ExpAddRecvOrdr 274
   ExpAddSendOrdr 350
   ExpAddTradProf 178
   ExpAsgAddrToDistList 105
   ExpAsgAddrToTradProf 107
   ExpAsgDistListToTradProf 160
   ExpAsgProjToCommProf 24
   ExpAsgRecvOrdrToDropBox 302
   ExpAsgSendOrdrToDropBox 378
   ExpCloseAdBk 90
   ExpCloseCInterface 3
   ExpCloseMbox 214
   ExpCloseMSys 475
   ExpCloseProj 16
   ExpDeleteAddr 109
   ExpDeleteCommProf 50
   ExpDeleteDistList 162
   ExpDeleteDropBox 242
   ExpDeleteIeLogon 68
   ExpDeleteProj 26
   ExpDeleteRecvOrdr 304
   ExpDeleteSendOrdr 380
   ExpDeleteSessRcpt 428
   ExpDeleteTradProf 189
Index

ExpDoPurgOrdr 504
ExpDoQuryOrdr 507
ExpDoRecvOrdr 518
ExpDoSendOrdr 536
ExpEndDropBox 481
ExpErrorExplanation 4
ExpListAddr 111
ExpListAddrOnDistList 121, 507
ExpListAddrOnTradProf 130
ExpListCommProf 52
ExpListDistList 164
ExpListDistListOnTradProf 169
ExpListDropBox 244
ExpListIeLogon 70
ExpListOrdersOnOrdrShlf 215
ExpListProj 27
ExpListProjOnCommProf 32
ExpListRcptsOnSessRcpt 221
ExpListRecvOrdr 306
ExpListRecvRcpt 439
ExpListSendOrdr 382
ExpListSendRcpt 460
ExpListSessRcpt 430
ExpListTradProf 191
ExpOpenAdBk 89
ExpOpenCInterface 2
ExpOpenMbox 213
ExpOpenMSys 474
ExpOpenProj 13
ExpProcDropBox 476
ExpRecoverSessRcpt 498
ExpReplaceAddr 139
ExpReplaceCommProf 58
ExpReplaceDistList 173
ExpReplaceDropBox 257
ExpReplaceField 7
ExpReplaceIeLogon 77
ExpReplaceProj 37
ExpReplaceRecvOrdr 322
ExpReplaceSendOrdr 399
ExpReplaceTradProf 200
ExpResetMem 8

ExpRetrieveErrText 82
ExpSizeOfOrdr 9
ExpSizeOfRcpt 10
ExpStartSess 487
ExpUsgAddrFromDistList 153
ExpUsgAddrFromTradProf 155
ExpUsgDistListFromTradProf 176
ExpUsgProjFromCommProf 43
ExpUsgRecvOrdrFromDropBox 348
ExpUsgSendOrdrFromDropBox 426

G
getting started 1
function
  ExpCloseCInterface 3
  ExpOpenCInterface 2

I
Information Exchange
  batch mode 474
  communicating with 474
  function
    ExpCloseMSys 475
    ExpDoPurgOrdr 504
    ExpDoQuryOrdr 507
    ExpDoRecvOrdr 518
    ExpDoSendOrdr 536
    ExpEndSess 481
    ExpOpenMSys 474
    ExpProcDropBox 476
    ExpRecoverSessRcpt 498
    ExpStartSess 487
  interactive mode 474
  interactive session commands 504
  listing users 70
  message system agent 474
input arguments
  ExpAddAddr 91
  ExpAddCommProj 45
  ExpAddDistList 157
Index

ExpAddDropBoxReqStruct 227
ExpAddIeLogonReqStruct 64
ExpAddProjReqStruct 18
ExpAddRecvOrdrReqStruct 278
ExpAddSendOrdrReqStruct 353
ExpAddTradProfReqStruct 180
ExpAsgAddrToDistListReqStruct 106
ExpAsgAddrToTradProfReqStruct 108
ExpAsgDistListToTradProfReqStruct 161
ExpAsgProjToCommProfReqStruct 25
ExpAsgRecvOrdrToDropBoxReqStruct 30
  3
ExpAsgSendOrdrToDropBoxReqStruct 37
  9
ExpCloseAdBkReqStruct 90
ExpCloseMboxReqStruct 214
ExpCloseMSysReqStruct 475
ExpCloseProjReqStruct 16
ExpDeleteAddrReqStruct 110
ExpDeleteCommProf 51
ExpDeleteDistListReqStruct 163
ExpDeleteDropBoxReqStruct 243
ExpDeleteIeLogonReqStruct 69
ExpDeleteProjReqStruct 26
ExpDeleteRecvOrdrReqStruct 305
ExpDeleteSendOrdrReqStruct 381
ExpDeleteSessRcptReqStruct 429
ExpDeleteTradProfReqStruct 190
ExpDoPurgOrdrReqStruct 505
ExpDoQuryOrdrReqStruct 508
ExpDoRecvOrdrReqStruct 521
ExpDoSendOrdrReqStruct 539
ExpListAddrsOnDistListReqStruct 122
ExpListAddrsOnTradProfReqStruct 131
ExpListCommProfReqStruct 53
ExpListDistListsOnTradProfReqStruct 170
ExpListDropBoxReqStruct 245
ExpListIeLogonReqStruct 71
ExpListOrdrsOnOrdrShlfReqStruct 217
ExpListProjReqStruct 28
ExpListProjOnCommProfReqStruct 32
ExpListRcptsOnSessRcptReqStruct 222
ExpListRecvOrdrReqStruct 307
ExpListRecvRcptReqStruct 440
ExpListReqStruct 112, 165
ExpListSendOrdrReqStruct 383
ExpListSendRcptReqStruct 461
ExpListSessRcptReqStruct 431
ExpListTradProfReqStruct 192
ExpOpenAdBkReqStruct 89
ExpOpenMbox 213
ExpOpenMSysReqStruct 474
ExpOpenProjReqStruct 14
ExpProcDropBoxReqStruct 477
ExpRecoverSessRcpt 499
ExpReplaceAddrReqStruct 141
ExpReplaceCommProf 59
ExpReplaceDistListReqStruct 174
ExpReplaceDropBoxReqStruct 259
ExpReplaceIeLogonReqStruct 78
ExpReplaceProjReqStruct 38
ExpReplaceRecvOrdrReqStruct 325
ExpReplaceSendOrdrReqStruct 402
ExpReplaceTradProfReqStruct 201
ExpRetrieveErrTextReqStruct 83
ExpStartSessReqStruct 489
ExpUsgAddrFromDistListReqStruct 154
ExpUsgAddrFromTradProfReqStruct 156
ExpUsgDistListFromTradProfReqStruct 17
  7
ExpUsgProjFromCommProfReqStruct 44
ExpUsgRecvOrdrFromDropBoxReqStruct 349
ExpUsgSendOrdrFromDropBoxReqStruct 427

interactive session commands 504
interactive sessions
end processing 481
file-level recovery 487
session-level recovery 487

J
Java class
ExpAddress 555
ExpAddressList extends LinkedList 557
ExpCommProfile 558
ExpCommProfileList extends LinkedList 559
ExpDBLister extends LinkedList 560
ExpDistListList extends LinkedList 564
ExpDistributionList 563
ExpDropBox 565
ExpDropBoxList extends LinkedList 567
ExpException extends Exception 568
ExpIELogon 569
ExpIELogonList extends LinkedList 570
ExpJni 571
ExpMailboxItem 572
ExpMailboxItemList extends LinkedList 573
ExpOrderList extends LinkedList 574
ExpProject 575
ExpProjectList extends LinkedList 576
ExpReceiptList extends LinkedList 581
ExpReceiveOrder 577
ExpReceiveOrderList extends LinkedList 582
ExpReceiveReceipt 579
ExpReceiveReceiptList extends LinkedList 583
ExpSendOrder 584
ExpSendOrderList extends LinkedList 587
ExpSendReceipt 586
ExpSendReceiptList extends LinkedList 588
ExpSession 591
ExpSessionReceipt 589
ExpSessionReceiptList extends LinkedList 590
ExpTradingProfile 593
ExpTradingProfileList extends LinkedList 594

logon IDs 63
adding 63
deleting 68
replacing information 77

M
mailbox 211, 213
closing 214
dropoff boxes 225
function

opening 213
receive receipts 439
send orders 350
send receipts 460
session receipts 428
writing the interface for 211
message system agent 475
miscellaneous tools, function
   ExpAddField  6
   ExpReplaceField  7
   ExpResetMem  8
   ExpSizeOfOrdr  9
   ExpSizeOfRcpt  10

O
   opening
      a mailbox  213
      message system agent  475

order shelf
   processing a receive order  518
   send order  536

output arguments
   ExpAddAddr  99
   ExpAddCommProf  48
   ExpAddDistList  159
   ExpAddDropBox  234
   ExpAddField  6
   ExpAddIeLogon  66
   ExpAddProj  21
   ExpAddRecvOrdr  290
   ExpAddSendOrdr  366
   ExpAddTradProf  185
   ExpCloseCInterface  3
   ExpDoPurgOrdr  506
   ExpDoQuryOrdr  509
   ExpDoRecvOrdr  534
   ExpDoSendOrdr  551
   ExpErrorExplanation  4, 5
   ExpListAddr  115
   ExpListAddrsOnDistList  124
   ExpListAddrsOnTradProf  133
   ExpListCommProf  56
   ExpListDistList  168
   ExpListDistListsOnTradProf  172
   ExpListDropBox  248
   ExpListIeLogon  74
   ExpListProj  29
   ExpListProjsOnCommProf  34
   ExpListRecvOrdr  310
   ExpListRecvRcpt  443
   ExpListSendOrdr  386
   ExpListSendRcpt  464
   ExpListSessRcpt  434
   ExpListTradProf  195
   ExpOpenCInterface  2
   ExpOpenProj  15
   ExpProcDropBox  478
   ExpReplaceAddr  147
   ExpReplaceCommProf  61
   ExpReplaceDistList  175
   ExpReplaceDropBox  266
   ExpReplaceField  7
   ExpReplaceIeLogon  80
   ExpReplaceProj  40
   ExpReplaceRecvOrdr  337
   ExpReplaceSendOrdr  415
   ExpReplaceTradProf  206
   ExpResetMem  8
   ExpRetrieveErrText  84
   ExpSessRcptRecovered  500
   ExpSizeOfOrdr  9
   ExpSizeOfRcpt  10

output field definitions
   ExpAddAddrRspStruct  101
   ExpAddCommProfRspStruct  48
   ExpAddDistListRspStruct  159
   ExpAddDropBoxRspStruct  236
   ExpAddIeLogonRspStruct  67
   ExpAddProjRspStruct  22
   ExpAddSendOrderReqStruct  369
   ExpAddTradProfRspStruct  186
   ExpDoRecvOrdrRspStruct  535
   ExpDoSendOrdrRspStruct  552
   ExpErrTextRetrieveRspStruct  84
   ExpIeLogonReplacedRspStruct  81
   ExpListAddrRspStruct  117
   ExpListAddrsOnDistListRspStruct  126
   ExpListAddrsOnTradProf  135
   ExpListCommProfRspStruct  56
   ExpListDistListRspStruct  168
ExpListDistListsOnTradProfRspStruct 17
ExpListDropBoxRspStruct 250
ExpListIeLogonRspStruct 75
ExpListOrdersOnOrdrShlfRspStruct 220
ExpListProjRspStruct 30
ExpListProjsOnCommProfRspStruct 35
ExpListReptsOnSessRcptRspStruct 224
ExpListRecvOrdrRspStruct 293, 313
ExpListRecvRcptRspStruct 447
ExpListSendOrdrRspStruct 389
ExpListSendRcptRspStruct 466
ExpListSessRcptRspStruct 435
ExpListTradProfRspStruct 196
ExpOpenProjRspStruct 15
ExpPurgOrdrDoneRspStruct 506
ExpQuryOrdrDoneRspStruct 511
ExpReplaceAddrRspStruct 149
ExpReplaceCommProfRspStruct 61
ExpReplaceDistListRspStruct 175
ExpReplaceDropBoxRspStruct 268
ExpReplaceProjRspStruct 41
ExpReplaceRecvOrdrRspStruct 339
ExpReplaceSendOrdrRspStruct 417
ExpReplaceTradProfRspStruct 207
ExpSessRcptRecovered 501

P
profiles

calendar function

ExpListProjsOnCommProf 32

trading profile 107

project password 14

projects

adding new 17

assigning to communications profile 24

closing 16

communications profile 26, 50

deleting 26

function

ExpAddCommProf 45

ExpAddIeLogon 63

ExpAddProj 17

ExpAsgProjToCommProf 24

ExpCloseProject 16

ExpDeleteCommProf 50

ExpDeleteIeLogon 68

ExpDeleteProj 26

ExpDoPurgOrdr 504

ExpDoQuryOrdr 507

ExpListCommProf 52

ExpListIeLogon 70

ExpListProj 27

ExpOpenProj 13

ExpReplaceCommProf 58

ExpReplaceIeLogon 77

ExpReplaceProj 37

ExpRetrieveErrText 82

ExpUsngProfFromCommProf 43

opening 13

password 14

replacing 37

unassigning from communications profile 43

public 555

R

receive orders 274, 304

adding 274

assigning 302, 348

dropoff boxes 242

replacing information about 322

unassigning 304

RECEIVE record 518

replacing

distribution lists 173

dropoff boxes 257

information about addresses 139

logon ID information 77

receiver orders 322

send orders 399

trading profiles 200
S
send orders
  adding 350
  assigning 378
  deleting 380
  dropoff boxes 242
  replacing 399
  unassigning 380
SENT record 536
session receipts, deleting 428
sessions, recovering 498

T
trading profiles
  adding 178
  deleting 189
  deleting a distribution list from 162
  deleting an address 109
  replacing 200
  unassigning
    distribution lists 189

U
unassigning
  receive orders 304
  send orders 380, 426
  trading profile addresses 189

W
Writing the interface
  for Information Exchange 473
  for projects 11
  for the address book 87
  for the mailbox 211