MIGRATION GUIDE

InterMail®

Version 4.0

Software.com™

THE INTERNET INFRASTRUCTURE COMPANY™

Document number: M-980601

Table of Contents

Preface

Welcome to InterMail!

The *InterMail Migration Guide* provides an overview of the migration process, instructions on how to migrate e-mail accounts from either Post.Office or sendmail, and advice on troubleshooting potential migration issues. This manual is intended for system administrators or technical support personnel who have a working knowledge of e-mail concepts and a detailed understanding of UNIX operating systems. It should be used as a complement to other InterMail documentation, specifically *The InterMail Reference Guide*.

While this guide is intended to give instructions where possible and to clearly define migration procedures, it cannot provide complete step-by-step migration instructions. This is because you will have requirements that are specific to your own provisioning system, the size of your database, available hardware, and so forth. The primary function of this guide is to serve as a template from which you can generate migration instructions that are specific to your own circumstances.

Overview of the Manual

The content of this manual is organized as follows:

- Chapter 1 provides an overview of the migration.
- Chapter 2 discusses the steps involved in the migration process.
- Chapter 3 explains methods for testing your migration.
- Chapter 4 covers migration troubleshooting procedures.
- Chapter 5 is a reference guide for Migration Toolbox Utilities.

Style and Conventions

To assist you in understanding the material presented in this manual, the following conventions have been observed:

Commands and configuration options are referenced by their proper names.

Environment variables (whose value is set at time of installation) are referenced with a preceding "$" (e.g., $spoolDir).

Commands (and other entries you might type) appear in `monospaced` type.

Variable names for elements within a command either appear between <angle brackets>.

Optional entries within a command appear in [square brackets].

Optional entries separated by a vertical bar (pipe) as in [option 1 | option 2] are exclusive—you can choose only one item from such a list.

Optional entries followed by an ellipsis (…) can be repeated. When an ellipsis follows a bracketed set, the expression within the brackets can be repeated.

{Curly braces} surround a list of options, one of which is required as an argument.

Boldface indicates literal input used in an example.

The process ID number (PID) is reported in the output files for `immigacctcreate` and `imboxmigrate` and is specified in the documentation as `<pid>`.

Questions and Comments

To suggest improvements or provide feedback on the content of this manual, send E-mail to InterMail.Manual@Software.com.

Legal Notices

Trademarks

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this documentation, and Software.com was aware of a trademark claim, the designations have been printed in initial caps or all caps.

InterMail and Software.com are trademarks of Software.com, Inc.

Licensing Agreement

The RSA Data Security, Inc. MD5 Message-Digest Algorithm

License is also granted to make and use derivative works provided that such works are identified as “derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm” in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

RSA Data Security, BSAFE

InterMail incorporates a derivative work of the BSAFE cryptographic toolkit, copyright 1992-1996, RSA Data Security, Inc. All rights reserved.

BSAFE is a trademark of RSA Data Security, Inc.

The RSA Public Key Cryptosystem is protected by U.S. Patent #4,405,829.

SSL Plus: SSL 3.0 Integration Suite Toolkit

InterMail incorporates a derivative work of the SSL Plus: SSL 3.0 Integration Suite Toolkit, copyright 1996, 1997 Consensus Development Corporation. SSL Plus: SSL 3.0 Integration Suite is a trademark of Consensus Development Corporation, which reserves all rights thereto.

Portions of the SSL Plus: SSL 3.0 Integration Suite Toolkit software are based on SSLRef(tm) 3.0, which is copyright (c)1996 by Netscape Communications Corporation. SSLRef(tm) was developed by Netscape Communications Corporation and Consensus Development Corporation.

The Regular Expression Routines

The Regular Expression Routines used in InterMail are © 1992-94 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.

The Regents of the University of California Copyright

InterMail includes software that is © 1990, 1993, 1994. The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Mike Olson.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Re-distributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Re-distributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU General Public License

Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.

675 Mass Ave., Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING,

DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as “you”.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave., Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

Oracle

Oracle Programs are the proprietary products of Oracle and are protected by copyright and other intellectual property laws. Customer acquires only the right to use Oracle Programs and does not acquire any rights, express or implied, in Oracle Programs or media containing Oracle Programs other than those specified by License. Oracle, or its licensor, shall at all times retain all rights, title, interest, including intellectual property rights, in Oracle Programs and media.

SmartHeap

Portions copyright 1991-1997 Compuware Corporation.

EMANATE

InterMail incorporates the EMANATE server as part of its monitoring functionality. Software.com licenses EMANATE pursuant to a license agreement with SNMP Research International, Incorporated. The copying and distribution of EMANATE is with the permission of SNMP Research International, Incorporated.

Apache Server License

Copyright (c) 1995-1997 The Apache Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   "This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/)."
4. The names "Apache Server" and "Apache Group" must not be used to endorse or promote products derived from this software without prior written permission.
5. Redistributions of any form whatsoever must retain the following acknowledgment:

   "This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/)."

THIS SOFTWARE IS PROVIDED BY THE APACHE GROUP ''AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE GROUP OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Group and was originally based on public domain software written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. For more information on the Apache Group and the Apache HTTP server project, please see <http://www.apache.org/>.

1

Migration Overview

Mailbox Migration allows you to move users' mailboxes from a "legacy" mail system (e.g., Post.Office or sendmail) into an InterMail system. The process of migrating mailboxes involves dumping the account database, creating accounts in the InterMail directory database, and migrating mailboxes to the InterMail Message Store Server (MSS).

Before beginning the actual migration process, you might want to look at a fairly typical migration strategy, such as the scenario presented in Chapter 5 of the *InterMail Operations Guide.* Also, if you are not yet familiar with the `imconfedit` utility, used to edit system configuration values, you should consult Chapter 3 of *The InterMail Operations Guide.* There are a number of these values you may need to set before beginning the actual migration process.

The remainder of this chapter covers the following topics:

- Migration as an incremental process
- Migration flow diagram

1.1 Migration as an Incremental Process

Typically, migration is performed in small increments. The first step is to create all accounts on the InterMail. After this step, mailboxes are migrated in user-definable increments. In this fashion, the administrator of an InterMail system can decide to move mailboxes when desired (i.e. time windows defined for system maintenance).

Creating all accounts in the destination InterMail system at one time and then migrating mailboxes on an as-needed basis has several benefits:

- The account creation procedures are executed off-line and do not affect system performance. When all accounts are successfully created in InterMail's Integrated Services Directory, DNS can be changed to point to the InterMail MTA and the Time to live (TTL) set to zero. After the MX record has been changed, all DNS requests will be pointed to the new MTA.
- If, due to an operator error, an insufficient amount of memory or disk space is encountered during the account creation process, system performance will not be affected because the process is off-line and the active mail system will still be the source Post.Office or sendmail.
- Issues which could possibly result in failure to create accounts, such as setting up forwarding or aliases, can be avoided by creating all accounts that have forwards associated with them in one batch. With this strategy, forwards and aliases are created properly because all accounts are created without forwards before any accounts that have aliases or forwards are processed.

1.2 Migration Flow Diagram

The chart below shows the operational flow in a typical UNIX InterMail migration process.

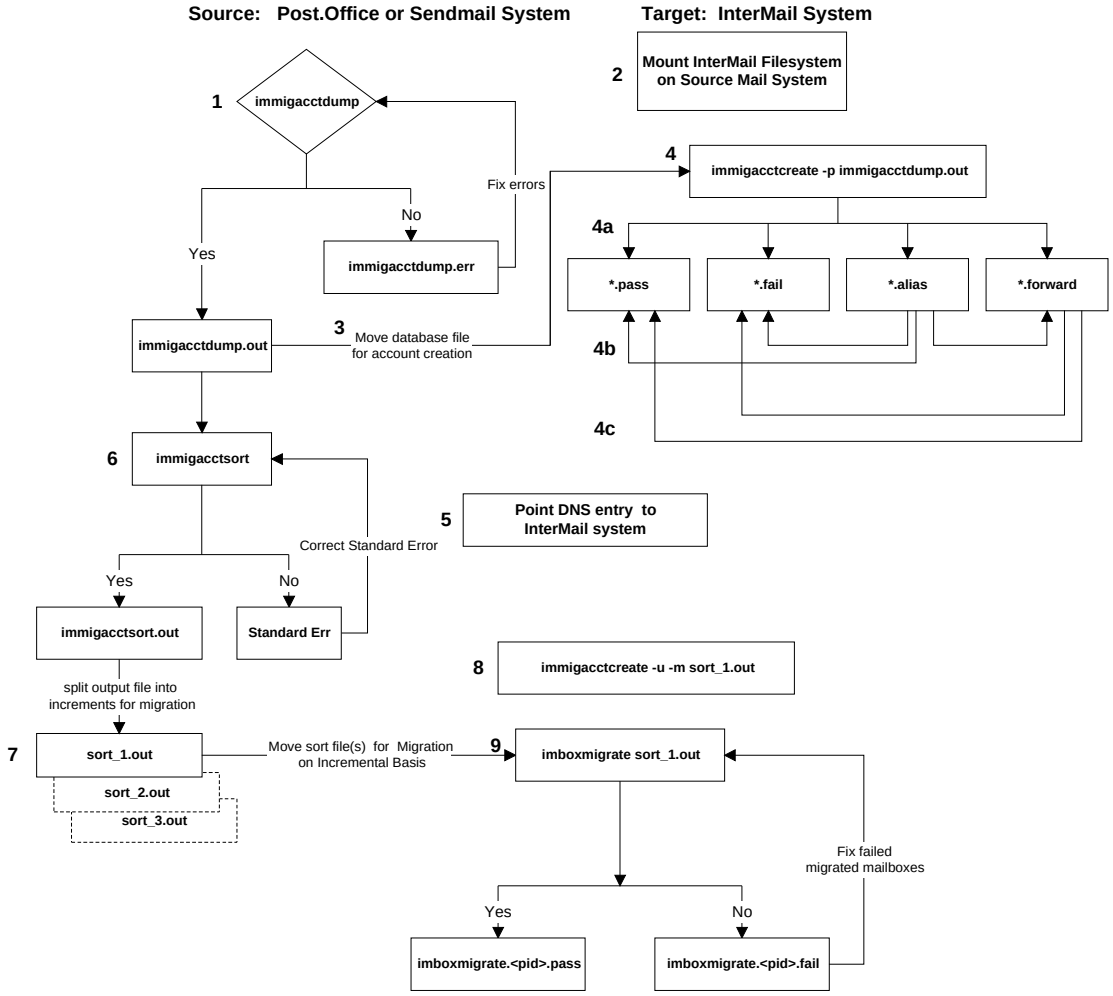

Figure 1 The Migration Process

1. Run `immigacctdump` on your existing account database from the legacy system, producing an `immigacctdump.out` file. If a `immigacctdump.err` file is produced, correct the errors and rerun `immigacctdump`.
2. Mount the InterMail file system on the Source Mail system.
3. Move correct `immigacctdump.out` file to the target InterMail system.
4. Run `immigacctcreate` on the `immigacctdump.out` file and the following occurs:
5. Accounts that are successful in PASS 1 are created and logged to `immigacctcreate.<pid>.create`. If accounts are not successful, error tags are logged in `immigacctcreate.<pid>.fail` file for these accounts. Also in PASS 1, an alias and forward file are created. These files have the aliases and forwards for the created accounts.

6. In PASS 2, the alias file is read and either an alias is added to an account created in Step 5, or an alias is added to the forward file. Aliases added to accounts that result in a successful account are logged to `immigcreate.<pid>.create`. If not successful, error tags are logged in `immigacctcreate.<pid>.fail` file for these accounts
7. In PASS 3, the forward file is read and added to the created accounts. As in Steps 5 (PASS 1) and 6 (PASS 2), if this results in a successful account, it is logged to `immigacctcreate.<pid>.create`. If accounts are not successful, error tags are logged in `immigacctcreate.<pid>.fail`.
8. After the complete pass and fail files have been created, check and fix accounts that have failed to be created.
9. After all accounts have been successfully been created on the InterMail system, point the DNS MX record entry to InterMail. All POP and SMTP requests will be directed to InterMail and will then proxy to the source mail system until such time as mailboxes for created accounts are present.
10. Take the `immigacctdump.out` file from Step 1 and run `immigacctsort`. If any errors are seen in the `immigacctsort` process, fix them and rerun.
11. Split the `immigacctsort.out` file into increments that can be used for migrating mailboxes (i.e. `immigacctsort_1.out`, `immigacctsort_1.out`, etc.).
12. Before migrating mailboxes, change account status of the mailboxes to be migrated to maintenance status. This will cause incoming mail to be queued and POP and SMTP sessions to be refused.
13. Run `imboxmigrate` on the first set of mailboxes to be migrated (`imacctsortsort_1.out`). If any mailboxes failed to migrate, check `imboxmigrate.<pid>.fail,` fix these errors, and rerun `imboxmigrate`.
14. After successful mailbox migration, change account status from maintenance to active. These accounts now have functional mailboxes and can receive mail.

2

The Migration Process

Migration is the process of moving existing user accounts and messages from a legacy mail system (such as Post.Office or sendmail) to an InterMail system. This process requires some initial system preparation. Once these preparations are complete, the actual migration procedure is done in several distinct steps. First, the existing legacy accounts are written to a text ("dump") file. Next, the dump file is sorted, and the resulting output is used to create a new account database in InterMail's Integrated Services Directory. Finally, all user messages are migrated from the legacy mail host onto one or more InterMail Message Store Servers.

Software.com provides a Migration Utility Toolbox to help with migration tasks.

Accounts are usually crated in "proxy" mode. This causes incoming messages to be proxied from InterMail on to the legacy system, allowing users to continue receiving mail with no visible impact on service.

Note: *To run in proxy mode you must first switch your DNS records from the legacy system to InterMail.*

During the mailbox migration phase, the status for new accounts is changed to "maintenance mode." This causes incoming messages to be deferred (saved to disk for later delivery) until mailbox migration is complete. POP and IMAP sessions are also denied during this phase, and users attempting to log on to the system are notified that a system enhancement is in progress. Once migration is complete, accounts are automatically switched back to active status. At this point, POP and IMAP sessions are again permitted, and all messages that were deferred during migration are delivered to their intended recipients.

Since the account creation phase should not have significant impact on service, you can probably run it any time. However, since the mailbox migration phase causes users' message to be deferred, you will probably want to run this procedure during off-peak hours, when the impact of deferred mail will be minimized. Since you can migrate mailboxes in batches (rather than all at once), you can perform this phase over a period of several days, during a series of "off-peak hour" windows.

This chapter offers a detailed description of all the steps involved in migration, but it is important to understand that your specific needs may require some modification of the procedures described.

The following sections cover:

- How Migration Works
- Multi-Pass Processing
- Preparing to Migrate Your Accounts
- The Migration Steps

2.1 How Migration Works

In simplest terms, account migration involves writing Post.Office or sendmail user information to a text ("dump") file, and then converting this into the Oracle database format used by InterMail's Integrated Services Directory (ISD). Mailbox and message migration is done in a separate phase, after all account information has been created. For a visual overview of the migration process, please refer to the Migration Flow diagram in Chapter 1.

The account creation phase of migration consists of writing legacy account data to a dump file, then using this file to create account information in InterMail's ISD.

The following table shows the account data parameters that will be created in the ISD. If your source accounts contain additional information, you will need to add this manually after migration is complete. Note that `po` indicates Post.Office data.

Field	Content
`Name`	The account user's name
`Account-Status`	Either `active`, `locked` or `suspended` for Post.Office. For sendmail, `active` only
`SMTP-Addresses`	A list of all the account's addresses and aliases
`Local-Delivery`	Whether the account delivers to a mailbox Options are `yes` or `no`
`POP-Login`	The login string used to access the account's mailbox
`Password-Type*`	Either `po-md5` or `UNIX`
`Password`	The account's password, encoded according to the password type
`Forward-To`	A list of addresses to which the account forwards
`Mailbox-Type`	Either `po`, `UNIX`, or `none`, specifying the type of mailbox to migrate
`Mailbox-Location*`	The directory (`po`) or filename (`UNIX`) where the mailbox resides
`Mailbox-Messages*`	The number of messages currently in the mailbox
`Mailbox-Size*`	The total number of bytes currently in the mailbox
`Mailbox-Idle-Time*`	The length of time since the mailbox was accessed, either by the user, or by the system when it last place a message in that user's mailbox
`Auto-Reply-Mode`	Either `vacation, reply, echo,` or `none`
`Auto-Reply-Message`	The text of an auto-reply message

*This information is used only for migration.

Section 2.5.5 explains how any of these parameters can be used as a key to sort and split the dump file in order to help manage the migration process.

Before reading about the steps involved in migrating accounts to InterMail from a legacy e-mail system, there are a few concepts and preliminary procedures you should know about.

2.1.1 Output and Intermediate Files

There are four important files to which data is incrementally written during the migration process. These are the:

- Pass file
- Fail file
- Alias file (during account creation phase)
- Forward file (during account creation phase)

Each of these four files is named according to the following conventions:

> `"immigacctcreate.<pid>.<pass/fail/alias/forward>"` where `<pid>` is the process id of the utility at runtime.

The two output files (`*.pass, *.fail`) and two intermediate files (`*.alias, *.forward`) are written out using the dump file format (blank line delimited account and alias data records). When necessary, certain of these output files can be re-submitted as input to `immigacctcreate` (the utility used to write account information to the ISD). Re-submitting output files during the migration process is explained later in this chapter.

Each execution of the `immigacctcreate` script adds success and failure tags to the files as appropriate. These tags accumulate on each data record.

2.2 Multi-Pass Processing

The `immigacctcreate` utility reads account information into the ISD in three passes. The three passes are necessary in order to ensure that account, alias and forwarding information are all created in the proper sequence. Although these passes are done automatically by `immigacctcreate`, the entire process requires some monitoring, as you will have to correct certain errors that occur as an expected part of migration.

The following sections describe the three migration passes in detail and briefly describe the types of errors that you can expect at each stage. These errors and some suggested strategies for correcting them are discussed in further detail in Chapter 4 of this manual.

Pass 1: Account Creation

During Pass 1, data records (either “account” or “alias”) are read in one at a time from the initial dump file. Processing is completed on one record before the next is read, and Pass 1 is not complete until the last data record is read in and processed.

Account and alias records are processed differently during Pass 1:

- **Account Records**

  For account records, all account parameters (as shown in section 2.1) are added to the ISD. The important exception to this is the `Forward-To` parameter.

  If the account record contains forwarding parameters then, after Pass 1 processing is complete, the record is written to the forward file. If the account record does not contain forwarding parameters, then it is written to the pass file. If record validation or database errors occur while processing the record, the record is written to the fail file with the appropriate error tags. Later, you will fix these errors and resubmit the fail file for another `immigacctcreate` run.

- **Alias Records (sendmail migrations only)**

  For alias records, the `immigacctcreate` utility either constructs a pseudo account record and processes it as it would an ordinary account record, or else leaves the record alone and simply passes it to the output alias file. The difference is determined by the contents of the `'Forward-To'` values.

  If there is a single `'Forward-To'` value, and the domain in the forwarding address is not a local domain, then this alias is a "Channel Alias." If there is more than one `'Forward-To'` value, then this is a `'Group-Alias,'` in which case a pseudo account record is created by adding the following account parameters to the input record:

  - `Name = POP-Login = Password = Primary SMTP Address,` with `'@'` replaced by `'_'`;
  - `Password-Type = clear;`
  - `Local-Delivery = off;` and
  - `Account-Status = 'P'` if proxy mode, `'M'` if maintenance mode, or `'A'` for active.

Note: *The account is created in the database in Pass 1, and the pseudo account record is added to the forward file for further processing in Pass 3.*

Pass 2: Alias Processing

During Pass 2, the alias file created during Pass 1 is used as input. Like Pass 1, `immigacctcreate` reads and process each record to completion before the next record is read. Pass 2 is complete after the final record in the alias file has been read and processed.

For each input record, the first value for the `'SMTP-Addresses'` key is tested to see if it represents an existing account. If it does represent an existing account, this is written to the forward file for processing in Pass 3. If the first value in `'SMTP-Addresses'` does not represent an existing account, then all the `'SMTP-Addresses'` listed are added as aliases in the `'Forward-To'` account. Note that in both of these cases, the `'Forward-To'` address is assumed to be an existing account . If the `'Forward-To'` address is not an existing local account, an error tag is added to the record when `immigacctcreate` attempts to add the alias in Pass 2. The record is then written to the fail file.

Pass 3: Forward Processing

During Pass 3, the forward file created during Pass 1 and Pass 2 is used as input. Like Pass 1 and Pass 2, `immigacctcreate` reads and processes each record in the input file to completion before the next record is read and processed. Pass 3 is complete after the final record in the forward file is read and processed.

The input data for forward processing contains both the account records (added in Pass 1) and alias records (added in Pass 2). For both account and alias records, only the forwarding information (listed under the `'Forward-To'` key) is added to the database. (This forwarding information is for the account represented by the first `'SMTP-Addresses'` value in each record.) For each individual `'Forward-To'` address an entry is added to the remote forward table

Successful processing of all forwards per record results in the record being written to the pass file (with the appropriate success tag added), while database or format errors cause the record to be written to the fail file.

Note: *Local forwards from Post.Office or sendmail are set up as 'Remote' forwards in InterMail.*

2.3 Preparing to Migrate Users

The following sections discuss some of the pre-migration tasks you may need to address before you begin the actual migration process.

InterMail Installation

In order to begin the migration process, InterMail must be properly installed, configured, and running on a remote system (a system other than your legacy mail system—e.g., Post.Office or sendmail.)

Environment Variables

The following variables need to be properly defined before migration of mailboxes:

1. InterMail Environment variable (`$INTERMAIL`)
2. Oracle home statement (`$ORACLE_HOME`)
3. ORACLE_DB_SERVICE

Database Connection Information

Either the config.db or an environment variable must contain the primary Oracle connection information. Check to make sure that the primary DB connection exists. Use the `primaryDbConnection` configuration key to define the primary DB connection:

For a detailed description of this configuration key, please see Chapter 11 of *The InterMail Reference Guide*.

Directories and Permissions

The `$INTERMAIL/config/config.db` database must exist.

The `$ORACLE_HOME/bin/sqlplus,/immigdbcontrol,` or `$INTERMAIL/bin/immigdbcontrol` directories must exist and be executable.

The `$INTERMAIL/config/<hostname>` file must exist, be readable, and contain the logical hostname.

immigacctdump on Source System

The `immigacctdump` utility must be installed on the Post.Office or sendmail source. This command reads the account database and produces an output file that can be used as input on the InterMail system. `immigacctdump` is part of the Migration Toolbox utility set.

Additional Configuration Settings

Before beginning the migration process, you should also verify the settings in the following configuration keys. If these keys are not correctly set, account creation in the Integrated Services Database (ISD) will not be possible.

- `/hostname/imdircacheserv/primaryDBconnection` and
- `/hostname/imdircacheserv/primaryDBuserInfo,` or
- `/hostname/common/primaryDBconnection` and
- `/hostname/common/primaryDBuserInfo,` or
- `/*/imdircacheserv/primaryDBconnection` and
- `/*/imdircacheserv/primaryDBuserInfo,` or
- `/*/common/primaryDBconnection` and
- `/*/common/primaryDBuserInfo`

The `primaryDbConnection` key defines the Oracle connection string for the ISD. This is the database that is queried by the Directory Cache server when information in the cache is considered out-of-date.

The `primaryDBuserInfo` key defined the Oracle username and password for ISD. This is the database that is queried by the Directory Cache server when information in the cache is considered out-of-date.

Optionally, you can also modify the value of the `imboxmigrateNumThreads` for memory optimization during the mailbox migration phase. In order to allocate proper memory resources, you may wish to change the default setting from 30 to some other value. If you are in the process of moving large mailboxes, you may wish to reduce the number of threads. If you are moving smaller mailboxes, you may wish to increase the number of threads in order to utilize CPUs and swap space more efficiently.

Note: *Threads can also be specified manually within the syntax of the* `imboxmigrate` *command.*

Creating the Necessary Domains and Sub-Domains

All local domains for all MTAs need to be specified, and your InterMail system must have at least one local domain specified. For a discussion of creating local domains on your InterMail system, see *The Integrated Services Directory Guide.*

The values in the Integrated Services Directory (ISD) must be coordinated with the data in your input file (the initial file submitted to `immigacctcreate`) or improper forwarding and database errors will result. All local domains must be defined in the ISD prior to running `immigacctcreate`.

You can create and edit domain information in the ISD using the `immigdbcontrol` utility. For a complete operational discussion of domain creation, refer to Chapter 3 of the *Integrated Services Directory Guide.*

Multiple Instances of config.db

If you install and configure multiple MTAs in your InterMail system (MTA1, MTA2, etc.), you should make sure to propagate any configuration changes to all the MTAs. You can do this using the InterMail Configuration Server (see Chapter 3 of *The InterMail Operations Guide.)*

Remove Test Accounts and/or Test Messages

You may wish to remove any test accounts and/or messages that were created during your initial system testing of InterMail.

Accessing the Integrated Services Directory

The MSS and client access machines need to be able to query the Integrated Services Directory, so you will need to make sure that all hosts have the correct settings for directory access. To do this, list the appropriate hosts using the `dirCacheHosts` configuration key

2.4 Proxy Mode and Migration

In order to provide continuous service to end users, you may wish to proxy incoming messages to your legacy system while users are being migrated. Section 2.5.6 contains a discussion of using proxy mode

There are two basic aspects to mail proxying:

- Accounts must be set to (or created in) proxy mode.
- InterMail must be configured to proxy mail to the proper remote host.

Section 2.5.6 in this chapter describes creating accounts in proxy mode. But before setting accounts for proxy, there are some configuration keys that must be set correctly.

To set InterMail to proxy incoming messages to another host during the account creation phase of migration, you will need to verify or change some configuration key settings using the `imconfedit` utility. If you are not yet familiar with using `imconfedit,` refer to Chapter 3 of the *InterMail Operations Guide.*

Using `imconfedit,` do the following:

1. Set the `popProxyHost` configuration key to the hostname of your source (POP3) system. For example: `/*/popserv/popProxyHost: [jupiter@accordance.com].`
2. Verify that the `popProxyPort` configuration key is properly set. This should be the POP port to use for the host specified in `popProxyHost`.
3. Set your relay host's source (SMTP) hostname using the `relayHost` configuration key.

Setting these keys will have the following effect:

If a user logs in and supplies and unknown POP username (expected if that user's account does not yet exist in the ISD), the InterMail POP Server will connect to the remote server you have defined in `popProxyHost`, using the port defined in `popProxyPort`. (By default this would be port 110, but you can specify a different port). The POP server then attempts a login with the username and password submitted by the user. If the login succeeds, the POP Server acts as a proxy during the remainder of the session.

In addition to handling POP connections, proxy mode allows SMTP relaying to be set. In relay mode, all mail destined for accounts that belong to a local domain known to InterMail (but which do not yet exist in the directory) are relayed to another SMTP server. This is useful for piecemeal migration from other mail systems, as described in the in the discussion on Proxy Methods in section 2.5.6.

The host to which mail should be relayed when addressed to unknown users at a local domain is set using the `relayHost` configuration key.

2.5 Migration Steps

Once you have completed your preparations, you should be ready to begin the actual migration process. The following list offers a brief outline of the steps involved in migration. These steps are then discussed in detail in the sections that follow.

1. Unpack the source media (the InterMail Migration Toolbox) onto the target machine—e.g., into the `$INTERMAIL/imail/bin` directory.
2. Mount the above directory on the source machine so that the toolbox utilities can be accessed from source machine as well as the target machine.
3. Create an output ("dump") file of your source account data using the `immigacctdump` utility.
4. Copy the output file to a temporary directory on the target machine (e.g., `$INTERMAIL/tmp`).
5. Sort the output file using the `immigacctsort` utility. This is an optional step you can take to break large output files up into smaller files in order to make migration more efficient. ***Note:*** *This step is optional*.
6. Run the `immigacctcreate` utility to create your user accounts in InterMail's Integrated Services Directory (ISD). Use the `-p` switch to put accounts in proxy mode. This will allow incoming messages to be proxied through to users in your source system while this phase is in process.
7. Fix any errors in the `*.fail` file and then use the result as the input for another `immigacctcreate` run.
8. Change your Internet DNS records so that they point to the new system. All mail for your users will now go there first. Proxying will allow you to send this mail on to your source system until the account creation phase is complete.
9. Start the mailbox migration process by running `imboxmigrate` on the target machine. All accounts should be in maintenance mode during this process. Once this process is complete, all migrated accounts are automatically switched back to "active" mode.
10. Check for any new messages that may have arrived at your source system and do another `imboxmigrate` pass to bring these over to your InterMail system.
11. If some of your source account information was not transferred during migration, you may need to add it manually.

2.5.1 Unpacking the Source Media

Software.com provides a Toolbox of utilities that will perform most of the migration tasks. Your first step in migration is to obtain the source media for the Toolbox and unpack the utilities. It is best to unpack the Migration Toolbox into a directory on your target (InterMail) system. Typically, this might be something like: `$INTERMAIL/imail/bin`, however the directory for the Migration Toolbox can have any name you like.

The Toolbox Utilities

The InterMail Migration Toolbox provides the utilities required to migrate Post.Office and sendmail accounts to an InterMail system. These tools are as follows:

- `immigacctdump`: This is the utility used to create a dump file from your source account information. This text file is then used as the initial input file for account migration.
- `immigacctsort`: This is the utility used to sort the "dump" file so that it can be split into smaller segments in order to facilitate migration. This process is optional, and whether or not you use it will depend to a large extent on the size of your existing account database.
- `immigacctcreate`: This is the utility used to create user accounts in the InterMail Integrated Services Directory. It's initial source file is the "dump" you create using `immigacctdump`.
- `imboxmigrate`: This is the utility used to migrate user mailboxes and messages from a Post.Office or sendmail system to an InterMail system. This part of the migration process should be done only after you have created your user accounts using `immigacctcreate`.
- `immigdbcontrol`: This is a utility used by `immigacctcreate` to write user account information in the Integrated Services Directory. You can also use this tool to edit account information manually should the need arise. It's use and syntax is virtually identical to that of `imdbcontrol`, which is discussed in the *Integrated Services Directory Guide.* Other than the command name itself, the syntax for `immigdbcontrol` is the same as for `imdbcontrol`.

2.5.2 Mounting The Target Directory On Your Source Machine

The purpose of this step is to allow free access to the Toolbox utilities from both the source and target machines, and also to allow mailbox and message migration from source to target machines. If you unpacked the Migration Toolbox utilities to `$INTERMAIL/imail/bin` on the target machine, then your mount point on the source machine must be the same.

Note: *Before account migration, you must be able to mount the file system(s) containing the actual mailboxes locally on the MSS host. If you cannot actually mount the file system, you will have to use a backup of the spool/mailbox areas and re-migrate/append the mailboxes again to gather any new mail received during the migration.*

2.5.3 Creating the Input File (Dumping the Source Database)

Now you are ready to create the initial input file. Essentially, this means producing a dump file of your source system's user information, which you will use as the initial raw material for creating accounts in InterMail's Integrated Services Directory (ISD).

The tool provided for writing the dump file is called `immigacctdump`. It's complete usage syntax will be somewhat different, depending upon whether you are migrating accounts from Post.Office or sendmail. This output file will contain the basic account information, as well any aliases you may have set up in your legacy system.

The `immigacctdump` utility searches the user database of an e-mail system and writes an account record for each account it finds. In addition to e-mail accounts, the output file will contain any other aliases that have been set up in the system. This output file is then used by the other migration utilities to create accounts and aliases, and to move mailboxes into InterMail. Some additional parameters that are written to the account record, (e.g., the mailbox size and number of messages) are useful for sorting (see `immigacctsort`) and to verify that the mailboxes have been moved successfully into InterMail.

The following table shows general syntax, as well as specific syntax for use with Post.Office and sendmail source systems.

Syntax	Description
General Options	
`-h`	Prints usage information.
`-t` *type*	Indicates type of system being migrated, either 'p' for Post.Office or 's' for sendmail
`-o` *file*	Creates file to collect output; by default, the output is written to standard output.
`-e` *file*	Creates file to collect error output; by default, the error output is written to standard error.
Post.Office-Specific Options	
`-c` file	Specifies a custom configuration file to use instead of the default of `/etc/post.office.conf`.
`-q` quota	Specifies mailbox quota to set up for users that don't have one (specified in kilobytes).
`-m` directory	Look in this directory for `mbox` files for users that have mail delivered to their UNIX mailbox; several locations can be specified and they will be searched in the same order as they appear on the command line; the '`~`' character is special and means to look in users' home directories for a file named `box`.

Sendmail-Specific Options	
`-d` domains	Set up all users' addresses in this domain; several domains can be specified to set up users in multiple domains.
`-m` directory	Look in this directory for users' box files; several locations can be specified and they will be searched in the same order as they appear on the command line; the '`~`' character is special and means to look in users' home directories for a file named `mbox`.
`-f` directory	Look in this directory for users' forwarding files; several locations can be specified and they will be searched for in the same order as they appear on the command line; the '`~`' character is special and means to look in users' home directories for a `.forward` file; otherwise a file named after the users' logins will be sought.
`-p` file	Specifies a custom password file to be used for migrating accounts; if unspecified, all current system accounts will be migrated; the file is expected to contain users' encrypted passwords.
`-s`	Specifies, when a custom password file is present, that the encrypted passwords are assumed to be in that file; this option causes `immigacctdump` to look up missing passwords in the shadow password database.
`-a` file	Indicates that aliases should be dumped in addition to accounts, and the named file is used as the source for alias definitions.
`-D` domain	Specify the domain used to complete short hostnames in addresses; by default, the completion domain is found in `/etc/resolv.conf`.
`-x`	Dump only accounts that have a mailbox file.

Dumping Accounts from Post.Office

When dumping a Post.Office account database, mailing lists and remote list subscriber accounts are skipped, since migration of this information is not currently supported. For the rest of the accounts, most of the information maps directly to InterMail account attributes, but there are a few differences. Since InterMail does not support UNIX or Program delivery at this time, accounts with this type of local delivery are converted to POP3 accounts when they are migrated. If an account does not have a POP login set in Post.Office, then the account's UNIX login is used as the InterMail POP login. However, this is only true if there is no conflict with another existing POP login. (`immigacctdump` also keeps track of the UNIX ids it assigns as POP logins, so it won't assign duplicates).

The `immigacctdump` command needs to be told where to find users' UNIX mailbox files. Typically these will all be located in a directory such as `/var/mail`, but some mail delivery programs can be customized to put them in a different directory, or even in users' home directories (as a file named `mbox`). To specify a list of directories to search for users' mailbox files, provide one or more `-m` options on the command line; they will be searched in the order provided, and the first mailbox file found will be reported in the output. Accounts that have both a UNIX and a POP3 mailbox in Post.Office will only have the POP3 mailbox written to the account record.

The accounts and aliases are written to the file `dump.out` and any errors are written to `dump.err`. When attempting to locate a mailbox file for users who have UNIX delivery enabled, `immigacctdump` first looks for a file named `/var/mail/`*`username`*. If that doesn't exist, `immigacctdump` looks for `~`*`username`*`/mbox`.

Example Dump From Post Office

A typical dump command for Post.Office might look like this:

```
immigacctdump -tp -m/var/mail -m~ -o immigacctdump.out -e
immigacctdump.err
```

The `-tp` portion of the command indicates a migration of type `(-t)` Post.Office `(p)`. The command syntax used here also specifies that data will be written to a file called `immigacctdump.out` (although you may use any output file name that you want, provided it is in the form `*.out`). Any errors encountered are written to `immigacctdump.err.` (Again, you can call the errors file anything you want, provided it has the form `*.err.`) If UNIX delivery is enabled for users, `immigacctdump` will look first for a file named `/var/mail/`*`username.`* If that does not exist, it will look for `~`*`username`*`/mbox` when attempting to locate a user's mailbox file.

Dumping Accounts From Sendmail

Migrating the accounts from a sendmail-based e-mail server requires gathering a lot of information scattered throughout the system. User accounts are usually defined in the `passwd` (password) database and aliases are set up in a file (`/etc/aliases` by default). Mailboxes are typically all located in a single directory, but some local delivery programs allow mailboxes to exist in users' home directories (as a file named `mbox`). Each user can optionally set up mail forwarding by creating a `.forward` file in his/her home directory.

Sometimes all users' forwarding files are stored in central directories, which have the same names as their UNIX logins. The source of each of these different pieces of information needs to be specified on the command line, since `immigacctdump` makes no attempt to guess. User accounts can either be gathered from the system (the current UNIX accounts), or may be specified in a custom file with the same format as `/etc/passwd`. The accounts in the file should contain encrypted passwords.

Note: *On systems that support shadow passwords, the encrypted passwords are still expected to be in the custom passwd file. To have* `immigacctdump` *look up missing passwords in the shadow password database, the* `-s` *option can be specified.*

To specify a list of directories to search for users' mailbox files, provide one or more `-m` options on the command line. These will be searched in the order listed, and the first mailbox file found will be reported in the output. The same is true of users' forwarding files. The directories to search are indicated with the `-f` option. A file containing aliases can be named by using the `-a` option.

As it searches for users' mailboxes, `immigacctdump` looks first for a file named `/var/mail/`*`username.`* If that doesn't exist, it looks for `~`*`username`*`/mbox`. It also looks for `.forward` files in users' home directories. In this sendmail example, each address and alias written to the output file would be set up in both the `software.com` and `hardware.com` domains. Domains must be specified here because, unlike Post.Office, sendmail does not recognize domains.

When a user has a forwarding file, it is parsed to find all of the deliveries. InterMail supported forwarding allows for the redirection of mail to other user by listing their addresses. If these addresses are not fully qualified, they are completed with the domain specified by the first `-d` option if the address has no domain, (e.g. `user`), or with the system domain specified in `/etc/resolv.conf`. Or else it might be the domain from the `-D` option, when the domain consists of only a short host name, e.g. `user@host`.

Note: *Each account data record in your source system must include at least one* `SMTP-Address` *value and an* `Account-Status` *value. If values for* `POP-Login`, `Password`, *or* `Password-Type` *are included, then all three must be present. If none of these are included, the* `POP-Login` *and* `Password` *default to the first* `SMTP-Address` *value (not including the domain). The* `Password-Type` *is also set to* `clear`*. Typically, such accounts have no local delivery, and in the case where the accounts are automatically generated via group alias or channel alias, the POP access is disabled by setting the account status to* `suspended`*. Finally, if an* `AutoReplyHost` or `Mailbox-Dst-Machine` *is specified and this machine differs from the host defined on the command line with the* `-h` *option, a warning is included in the terminal output.*

Piping to a program, delivering to a file, and reading other files containing more forwarding information (using sendmail's `:include:` directive) are not supported, so an error message is printed when any of these is encountered. The one exception to this is if the user is piping mail to the *vacation* program. In that case, the account is set up to deliver vacation messages, and no error message is reported. The user's vacation message is plucked from a file called `.vacation.msg` found in the user's home directory.

Similar restrictions are applied to the alias file (if one was specified).

The following are ignored:

- All pipes to programs
- Deliveries to files
- `:include:` files
- Local delivery aliases of the form `\username`.

Example Dump for Sendmail

A typical dump command for sendmail might look like this:

```
immigacctdump -ts -d software.com -d hardware.com -m/var/mail -m~ -f~
-a/etc/aliases -o immigacctdump.out -e immigacctdump.err
```

As in the Post.Office example, a type must be specified (in this example, it is `-ts` for type `(-t)` sendmail `(s)`. The accounts and aliases will be written to the file `immigacctdump.out` and any errors will be written to `immigacctdump.err`. Aliases will be scanned from the `/etc/aliases` file.

2.5.4 Copying the Output File to an InterMail Directory

The next step in migration consists of copying the output dump file to a host on your fully installed InterMail system.

2.5.5 Sorting the Account Database

The dump file you created using the `immigacctdump` utility will (by default) be sorted in the same order as were the accounts on your source system. In some cases, you may simply wish to leave the file as is, however there are some situations where you might want to re-sort the dump file in order make the rest of your migration more efficient. For example, you may want to sort the file by number of messages stored for each account. This would allow you to split the dump file at some point that makes sense to you, so that you could migrate just those accounts with very few (or very many) messages.

Let's say that you decide that those accounts with that currently have no stored messages should be created first. That might make sense in that these accounts can be migrated very quickly and at any time, because you will not need to do the mailbox migration phase.

Note: *No mailbox migration is needed here because there are no current messages to store. The first time these users receive a message on the InterMail system, the system would automatically create a mailbox for them and store the message there. This, of course, assumes that there are accounts for these users in the Directory.*

In this case, you would split the dump file into two files, one containing accounts with 0 stored messages, and the other containing accounts with one or more. You can now create these two batches of accounts separately in the ISD, as described in section 2.5.6. (Two files is just an arbitrary number here, since you split the initial file into any number of files, depending upon your needs.)

The `immigacctsort` command uses the following syntax:

```
immigacctsort [-i <file>] [-o <file>] <sort_param>
```

where `-i` is followed by the name of the input file and `-o` is followed by the name of an output file. The input file is then sorted according to the argument for `<sort_param>.` If you wanted to sort the file by number of messages, the argument for `<sort_param>.` would be `Mailbox-Messages.`

Refer to the following table for an explanation of syntax.

Syntax	Description
`-h`	Prints this usage information.
`-i` *file*	File to use as input; by default, the input is read from standard input.
`-o` *file*	File to collect output; by default, the output is written to standard output.
`sort_param`	A list of parameters to compare when sorting the input, such as Mailbox-Size. Sorting will be in alphabetical or numerical order, depending upon the parameter.

The following syntax would sort a file so as to group accounts based on whether or not these users have local delivery turned on. (Users who have local delivery turned off are not presently receiving mail at a local mail box, so again there'd be no mailbox migration phase to worry about for these accounts):

```
immigacctsort -i immigacctcreate.pass -o immigacctsort.out Local-
Delivery
```

In the preceding example, an `immigacctcreate.pass` file (see Section 2.1.1) is used as the input (`-i`) file. The output (`-o`) file is called `immigacctsort.out`, and the operation will sort according to the Local-Delivery sort key. The following is an example of what the `immigacctsort.out` file might look like.

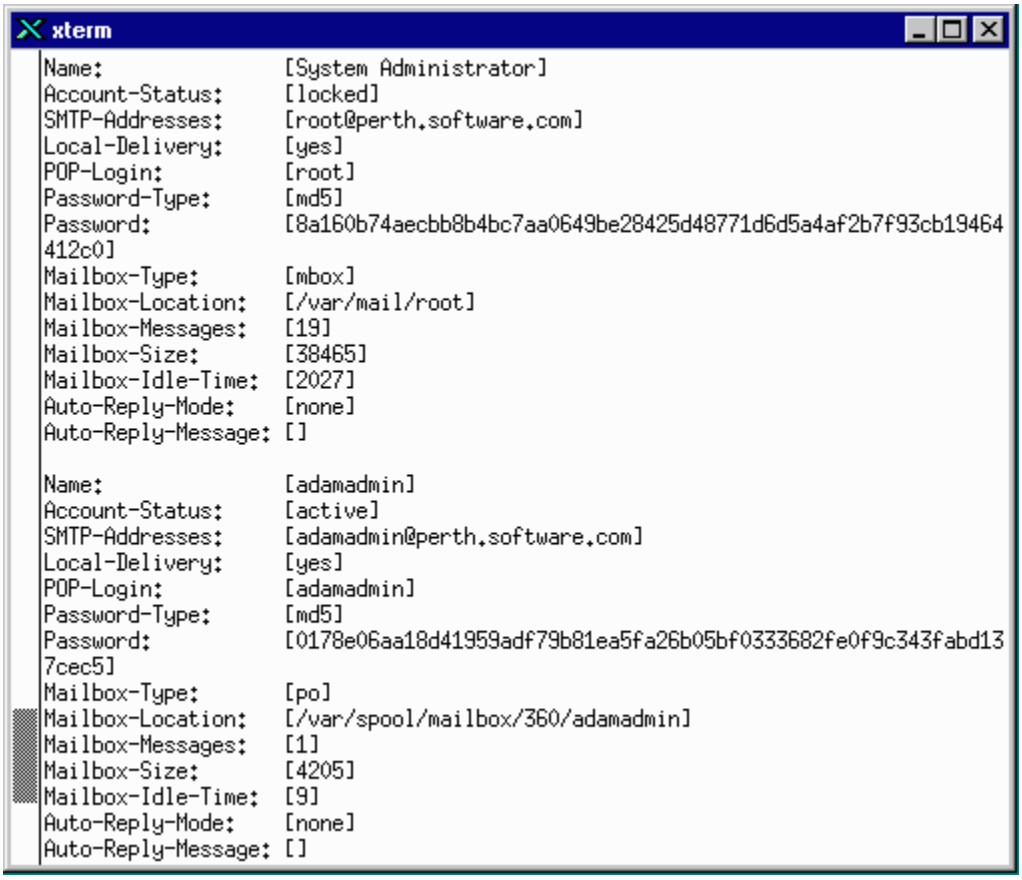

Figure 2 Sample immigacctsort.out File

This figure shows a sample of the sorted account records (including values) in `immigacctsort.out`. In this example, the records are sorted according to `Local-Delivery`. You can then split the `immigacctsort.out` file into two files at the point where all `the Local-Delivery` field values switch from `yes` to `no`. You can then use the two files for two separate account creation runs, performing them at times that make sense to you.

Determining Whether Sorting Will Be Useful

Whether or not sorting the initial input file will be useful depends upon the target and source systems involved in your migration. If the account database is relatively small (around 50k accounts) and a possible disruption in operations is not a major issue, you may decide to create all your accounts at the same time, in which case there'd be no need to sort and split the file.

Note: *If your legacy e-mail system has accounts that use forwarding, you* must *process all the forwarding accounts at the same time. However, this may have some performance implications. On multiprocessor machines, accounts can be created much faster by running several instances of* `immgacctcreate` *on subsets of the dump file. If forwarding is used and the dump file is split into subsets for use with* `immgacctcreate`, *then the account data subset that is being processed should have all of the data records which include* `Forward-To` *parameters.*

Sort Parameters

The output file produced by `immigacctdump` can be sorted based on one any combination of the account parameters (sort keys) listed below.

`Name`	The name of the account. Usually this would be the user's real name.
`Account-Status`	Either `active, locked` or `suspended`
`SMTP-Addresses`	A list of all the account's addresses and aliases
`Local-Delivery`	Whether the account delivers to a mailbox (`yes` or `no`)
`POP-Login`	The login string used to access the account's mailbox
`Password-Type`	Either `md5-po` or `UNIX` mode
`Password`	The account's password, encoded according to type
`Forward-To`	A list of addresses to which the account forwards
`Mailbox-Type`	Either `po`, `UNIX`, or `none`, specifying the type of mailbox to migrate
`Mailbox-Location`	The directory (po) or filename (UNIX) where the mailbox resides
`Mailbox-Messages`	The number of messages in the mailbox
`Mailbox-Size`	The total number of bites in the mailbox
`Mailbox-Idle-Time`	The length of time since the mailbox was last accessed
`Mailbox-Quota`	The quota to set on the mailbox when it is migrated to InterMail
`Auto-Reply-Mode`	Either `vacation, reply, echo,` or `none`
`Auto-Reply-Message`	The text of the auto-reply message

An account will sort alphabetically or numerically based on the specified parameter (sort key). Accounts that do not have a value for the selected sort key will sort after all the accounts that do have values. You can use multiple parameters for a sort, in which case subsequent keys (after the first) are compared only where all the previously stated parameters yield identical values.

If the value of a sort parameter in an account consists of only digits, it is assumed to be numeric and is padded on the left with enough spaces to make the total length ten digits. This is done so that numeric values will always sort in the correct order. For example "9" would normally sort after "10" in a text file, but if numeric values are padded with spaces (e.g., " 9") then "9" will sort before "10."

Sort keys are treated in a case-insensitive manner and only need to be specified up to a unique prefix, meaning that POP-Login and pop would be viewed identically.

2.5.6 Creating Accounts

Having created and sorted your initial input file (and changed your DNS records), you should now be ready to start creating InterMail accounts in the Integrated Services Directory. This is done using the `immigacctcreate` utility.

Note: *Remember that, before you can begin account migration, you must be certain that all the domains for the accounts to be migrated exist in the InterMail ISD.*

The `immigacctcreate` utility also has other uses, such as setting a batch of accounts to maintenance mode prior to the mailbox migration phase, this discussion chiefly concerns using the utility for account creation. The command has the following syntax:

```
immigacctcreate -h <host> -a <file> [-m] [-u] [-o file]
```

The following table provides an explanation of the syntax for `immigacctcreate`.

Syntax	Description
`-help`	Prints this usage information.
`-h` *host*	MSS host name.
`-a` *file*	Specifies a file containing input account data; format matches `immigacctdump` output format.
`-m`	Forces account status to 'maintenance' as accounts are created or updated regardless of the account status specified in the input data; default 'off.'
`-u`	Forces 'update' rather than 'create.' Accounts are updated with the input account data; an error occurs if the account does not already exist; default 'off'.
`-e` *file*	file to collect STDERR; contains status output and progress indications
`MSS-x` host	POP Proxy host
`-r` host	Auto-reply host (SMTP relay/proxy host)
`-p`	Proxy mode

The `immigacctcreate` command also produces the `immigacctcreate.<pid>.pass` and `immigacctcreate.<pid>.fail` files, which are described in section 2.1.1. You will need to check the fail file and correct any errors until you have the complete set of mailboxes you want to migrate in the pass files.

```
immigacctcreate -h <mss-host> -p -x <pop-proxy-host> -r <smtp-
proxy/relay-host> -a <output-filename>
```

Once the output file resides on the InterMail host, use the following command syntax to create the accounts:

```
immigacctcreate -h mss1.software.com -a immigacctdump.out -p -x
source-host -r source-host-name -o pass
```

This example sets up the MSS host name `(-h)` on which mailboxes will be created during the mailbox migration phase(e.g., `mss1.software.com`). Also specified is the input file `(-a)` that will be used to create accounts `(dump.out)`

In this example, accounts are created in proxy mode (`-p`). Since no host is specified in the example shown (`no -x`), `-p` defaults to the setting in the `popProxyHost` configuration key. You can specify an MSS host (`-h`), as well as an auto-reply host (`-r`). The `-o` *`pass`* and `-e` *`fail`* switches specify the creation of files for passed and failed account records.

Update Mode

Note that `immigacctcreate` can also be used with the update mode (`-u`) switch to modify account parameters. Only a subset of account parameters are updated. No `SMTP alias`, `Forwarding`, `auto reply`, or `local delivery` settings are affected, so these parameters in the input file are essentially ignored. The main purpose of using update mode is to change the account status values for an entire group of accounts—as for example, when you want to change accounts from proxy to maintenance mode prior to the mailbox migration phase.

***Note**: The value of `Account-Status` (written to the output pass and fail files) remains whatever was stipulated in the original dump file, so running `immigacctcreate` in update mode will simply return an account to this setting. Usually, this would be `active`, but it could also be `suspended`, etc. This means that suspended accounts would not be reset to active mode.*

Creating Accounts In Proxy Mode

Proxy mode is commonly used to proxy mail from one MSS host and/or relay (MTA) host to another. As previously described, this mode is useful during the account creation phase of migration. Section 2.4 discussed configuring global proxy settings for InterMail. This section discusses creating accounts in proxy mode during the migration process. These methods allow a bit more flexibility in that they allow you proxy mail to multiple hosts, rather than just a single, globally defined proxy host.

Proxy mode is invoked using the `-p` command line switch. This mode causes the mail store address in the ISD to be used as the host address for POP proxy connections

Proxy Methods

There are two proxy methods you can use during migration. Considerations for employing these methods require a few procedures prior to using `immigacctcreate`:

- **Known User Proxy Mode**

  Known User Proxy Mode is useful when there are several source POP or MTA proxy hosts contained in your source system and you do not want to overburden one particular host or set of hosts. To run in Known User Proxy mode, do the following:

1. Mount your source system (Post.Office or sendmail) to the new InterMail system, as described in section 2.5.2.
2. Either:
   - Run `immigacctdump` to dump account file and then run `immigacctsort` on any account key. The idea is to break the file up in some way so that one batch of accounts can be proxied to the first host and another batch to the second (or third, etc.).
   - Run `immigacctdump` and edit the resulting output manually.

3. Now, run `immigacctcreate` in iterations, specifying the POP, MTA and relay hosts in the following manner:

   ```
   immigacctcreate -h <MSS-host> -a <dump_1> -p -x hostA -r hostA
   immigacctcreate -h <MSS-host> -a <dump_2> -p -x hostB -r hostB
   ```

- **Unknown User Proxy Mode**

  Unknown User Proxy Mode is useful when proxy mode is specified on a host-wide basis and values are assigned to the `popProxyHost` and `relayHost` configuration keys. To run in Unknown User Proxy Mode, do the following:

1. Add the `/*/popserv/popProxyHost` key to the new MSS-host.
2. Add the `/*/mta/relayHost` to the new MSS-host.
3. Mount your source system (Post.Office or sendmail) to the new InterMail system, as described in section 2.5.2.
4. Run an `immigacctdump` of your source system.
5. Run `immigacctcreate` on the dump file using the following syntax:

   ```
   immigacctcreate -h <MSS-host> -a <dump file> -p
   ```

   If accounts are added in active mode, global proxy has no effect. Global (unknown user) proxy is useful only if you create accounts piecemeal.

You can also insert a fully qualified domain name as the mss host information for any individual account. If this account is set to proxy mode, then the mss host value is interpreted as a POP proxy host value, which overrides the setting in the `popProxyHost` configuration key. In this way, you can use the `popProxyHost` setting as a global default proxy host, but still set individual accounts to proxy to a different host.

Maintenance Mode

The `immigacctcreate` utility can also be used to set accounts to maintenance mode. Accounts must be switched from proxy to maintenance mode before you begin migrating mailboxes and messages. While a user's account is in maintenance mode, no POP or IMAP access to this account is permitted. Also, mail for this account will be deferred (queued) for future delivery until the migration process is complete.

When `imboxmigrate` has completed mailbox migration, it automatically switches the accounts from maintenance mode to "active" status. At this point, mail that has been queuing for "migrating users" is delivered to their mailboxes.

Note: *Accounts must be set to maintenance mode before you begin the mailbox migration phase.*

2.5.7 Fixing Errors in the Fail File

As explained in section 2.2, the account creation process involves three distinct passes through the account data. During this process, there are certain points where you can expect errors to occur for specific accounts. In most cases, this does not indicate a serious difficulty, but it will require you to check the error tags in the fail file and manually clean up the account problems described by the tags.

2.5.8 Changing Your DNS Records

Up until now, mail for users has continued to arrive at your legacy system. But once all your accounts are in the ISD, you can safely reset your DNS records to point at the InterMail MTA host, rather than at your legacy system. Since your accounts are in proxy mode, users mail will first arrive at the InterMail MTA, then be proxied on to the legacy system for actual delivery. When you are ready to migrate the first batch of mailboxes, you will switch these accounts from proxy to maintenance mode and begin mailbox migration for this batch. When this batch is done, these accounts are switched to active mode and can begin receiving mail on the InterMail system.

Note: *Make sure that Directory Cache Servers are running before switching your DNS records.*

2.5.9 Migrating Your Mailboxes

Once you've completed the preceding steps, you can begin migrating users' mailboxes from your source system to InterMail. To do this, you will run `imboxmigrate` on the target (InterMail) system. The `imboxmigrate` utility checks the directory for the correct host and migrates the mailboxes there. (This is the host that you specified using the `-h` switch when you ran `immigacctcreate`.)

Note: *"Mailboxes" are only migrated where there are existing messages for a user. If there are no messages for that user, no mailbox creation takes place. Instead, a mailbox for this user is created "on-the-fly" the first time the user receives a message on the InterMail system.*

The `imboxmigrate` utility migrates user mailboxes (plus all stored messages) from Berkeley mailbox format or from Post.Office format to an InterMail Message Store Server. The `<batch_file_name>` parameter specifies a dump file that contains the entries and specifies the users whose mailboxes will be moved.

In Post.Office mailbox migrations, several steps must be taken to provide access to the Post.Office accounts in InterMail:

- There must be an `/etc/post.office.conf` file from the source host on the InterMail host with the appropriate mount point listed in the `MailboxDir` and `PostOffice` fields.
- There must be a user on the InterMail host with the same `username` (usually `mta`) as in the `MailUserName` field in `/etc/post.office.conf.` That user must also have the same UID as the Post.Office host.
- You should also create a tape dump or tar file of mail that can be copied to the target InterMail system.

Note: *When you run* `imboxmigrate` *on an MSS host for the first time, you should first create an imail mailbox for the imail account. This is because* `imboxmigrate` *needs an imail mailbox in order to create the target mailboxes used in migration.*

Mailbox Migration Syntax

In the following example, let's assume that the `immigacctsort.out` file has been split into several files (`*sort1.out, *sort2.out, *sort3.out`) based on the sort key applied to it. Use the following syntax for `imboxmigrate`:

```
imboxmigrate [-h][-append | -force | -create |- forewarn][-compare |
-inbox | -nocompare] [-threads int] sort filename
```

Refer to the following table for an explanation of syntax:

Syntax	Description
`-h`	Prints this usage information.
`-append`	Use this switch to add (append) messages from a source system to mailboxes already existing on the target system. This will fail if a mailbox does not exist on the target system.
`-force`	Creates a message store if mailbox is not present.
`-create`	Use this switch to create mailboxes on the target and add all existing messages from the source system.
`-forewarn`	Creates a message store if it is not present, but a warning message will be logged.
`-compare`	Sets a validation phase to compare the original message with the stored message after the messages have been migrated.
`-inbox`	Compares only the messages in INBOX (this requires `-compare`).
`-nocompare`	Turns off the validation phase.
`-threads int`	Sets number of threads to use while moving mailboxes. Each thread moves one mailbox at a time. This specification will override the specification set using the `imboxmigrateNumThreads` configuration key. If there is no specification, the default is 30 The maximum recommended number to set with `imboxmigrateNumThreads` is 2000.

The following example shows how the `imboxmigrate` command could be used to migrate the first set of mailboxes.

```
imboxmigrate immigacctsort1.out
```

Note: *`imboxmigrate` with no options sets `-create` and `-compare` as defaults and threads default to 30.*

In this command, mailboxes are created and the command will exit with an error condition if a mailbox for an account specified already exists. After messages in the mailbox have been successfully moved, the original messages are compared with the stored messages. Finally, the input file (`immigacctsort1.out`) used in the process is specified.

Validation Phase

After the messages have been migrated, you can run a validation phase to compare the original messages with the stored messages.

```
-compare | -inbox | -nocompare
```

The `-compare` option specifies that all original messages are compared with all the newly stored messages. The `-inbox` option specifies that only messages in the "inbox" will be compared. The `-nocompare` option turns off the validation phase. The default setting is `-compare`.

If the validation phase fails, `imboxmigrate` will *not* delete the copied messages or otherwise restore the destination mailbox to its initial state.

Pass Files

The `imboxmigrate` command creates a pass-file containing the entries that were successfully moved (and compared if specified) and a fail-file containing those entries that were not successfully moved, or if the comparison phase for that account failed. Each of the fail-file entries is appended with information that specifies the reason for failure as best as could be determined by `imboxmigrate.` The filename is:

```
imboxmigrate.<Process-id>.[ pass | fail ]
```

Note: *For a discussion of the fail file and errors that can occur during the mailbox migration phase, see Chapter 4.*

2.5.10 Checking For New Messages

If you used proxy mode at any time during the migration process, you will need to make sure that any additional messages that may have arrived at the proxy host since you began are brought over to the InterMail system. To do this, you can run another `imboxmigrate` command using the `-append` switch described in the table in section 2.5.8. This will search for any new messages on the source system and bring them over to the appropriate mailboxes on the InterMail system.

2.5.11 Updating New Accounts in the ISD

The `immigacctdump` utility creates the input file using all the account information specified in section 2.1. However, it is possible that your source user database contains some other account information that cannot be migrated using `immigacctdump` and `immigacctcreate`.

If that is the case, you may need to edit the account information in InterMail's ISD directly. The `immigdbcontrol` utility allows you to manually add and modify account data in the ISD. Chapter 5 of this manual contains the basic syntax and some other reference material for using this tool. A complete operational discussion of editing account information can be found in Chapter 3 of the *Integrated Services Directory Guide.*

Note: `immigdbcontrol` *is the functional equivalent of* `imdbcontrol`*, the tool discussed in Chapter 3 of the* Integrated Services Directory Guide.

3

Testing the Migration Process

This chapter offers some basic tips on testing the migration process. Essentially, testing means doing a "dry-run" on some account data. This will give you a chance to practice with the migration utilities, get a feel for how they work, and see what types of errors might occur during the various phases of migration. Once you feel comfortable with the process, you can begin migrating your actual account data to InterMail.

The discussions in this chapter assume that you have already familiarized yourself with the migration steps, tools, files and terminology discussed in Chapter 2 of this manual. As you perform your tests, you can use the migration steps as a guide. You should also use Chapter 4 as a reference for understanding the various types of errors that may occur during migration.

To run a migration test, you will first need some account data to transfer from your source system to InterMail. Basically, there are a couple of way to approach this. You can either create some fictitious accounts (and messages) on your source system and try migrating these, or you can use a subset of your real account data. The second method has an advantage in that it will save you the effort of creating fictional accounts and messages for their mailboxes. Using actual data will also create a migration scenario that is a bit more like "the real thing."

The remainder of this chapter covers the following topics:

- Creating a Test Database
- Creating Test Accounts
- Testing Mailbox Migration

3.1 Creating a Test Database

One advantage of creating a relatively small account database for your test is that you can move through the entire migration process much more quickly. As mentioned earlier, you can create a fictional database, populate its mailboxes with messages, and then use this data for your practice run. For sake of simplicity, though, we'll assume you've decided to choose the second method and use a subset of your real account data for practice.

Start by copying somewhere between 100 and 200 accounts and mailboxes from your source database. Put these in a unique set of directories, and use the accounts to crate a dump file for your practice run. (The section on using `immigacctdump` in Chapter 2 describes the process of creating a dump file of account information.)

When you run your test, you should try to "stress" the system as much as possible. Migrating accounts with no messages or only a few messages is fairly trivial, and you can certainly try testing some of these. But the real test is in migrating accounts with dozens (or even hundreds) of messages in their mailboxes. So, when you choose a set of accounts for your dry-run, try to include a fair number that have a lot of messages. At the very least, you should try for a representative sampling of your entire source account database.

3.2 Creating Test Accounts

Once you've created an account database to practice on, the next step will be to run the `immigacctdump` utility to produce a text file which will serve as the source for your InterMail account data.

Once you have the initial "dump" file, there are a couple of things you can try:

- Run `immigacctcreate` on the entire dump file

  This is the simplest test. You just run the `immigacctcreate` utility on the entire `immigacctdump.out` file, and proceed as described in Chapter 2.

- Split the dump file and then run `immigacctcreate` on the resulting files

  In this test, you would first sort the dump file using the `immigacctsort` utility, split the initial file into two or more files, then do an `immigacctcreate` run on each one.

Since you will be practicing with a small account database (and since the whole point of this is to get some practice) there is no reason why you can't try both scenarios. Do the `immigacctcreate` run on the entire file first, since this is simplest method. You can then do one of two things: either continue the rest of the migration steps, or else wipe the account database you're created in the ISD, and then create a new one from a dump file you've split into two or more files.

Some strategies for sorting and splitting a dump file are discussed in Chapter 2, and these practice runs can be a good opportunity to try a couple of different strategies to see what actually makes sense for you.

***Note**: You should pay particular attention to the discussion in Chapter 2 that describes creating accounts that have forwards associated with them. Doing a test* `immigacctcreate` *run on these and verifying the results is one of the most important tests you can do.*

Proxy Mode

Chapter 2 offers a discussion of proxy mode, and also explains the importance of creating accounts in proxy mode so that incoming messages can be directed to your source system while the account creation phase of migration is in progress. Testing proxy mode will require you to set up a POP Proxy Host.

Create the accounts in the ISD using the `immigacctcreate` utility with: the`"-x"` flag for POP Proxy and the `"-r"` flag for the individual SMTP Relay Host.

Testing Pop/SMTP Proxy

You should test to make sure that POP Proxying takes place properly. You can do this by using a `tail -f` command, which lets you set up a console session to see proxy transactions as they occur in real time.

Fixing Errors

There are specific errors that can occur during any phase of migration. Many of these errors are no cause for alarm and are, in fact, an expected and important part of the process. Errors are written to the `*.fail` files described in Chapter 2. The larger the number of accounts in your test data, the more likely it that you will have some account creation errors.

One important migration step consists of fixing these errors and then submitting the repaired data as input for another `immigacctcreate` run. Chapter 4 offers a list of typical errors that may occur during the account creation phase of migration.

3.3 Testing Mailbox Migration

After creating your test accounts in the Integrated Services Directory, you will need to test the final phase of the migration process—mailbox and message migration. As described in Chapter 2, the migration phase is done using the `imboxmigrate` utility.

Make sure that all accounts are switched to maintenance mode for this phase. With accounts in maintenance mode, all incoming mail for these users should be queued, and POP and IMAP access for these accounts should be prohibited. You can verify that this is happening by trying to POP a mailbox from the user end, and by sending some messages during the mailbox migration phase to see if they queue as expected. When the migration phase is complete, `imboxmigrate` should return accounts to “active” mode. The exception to his is if the accounts were in some other mode (such as “suspended”) to begin with. If so, they should be returned to their former state—i.e., these accounts should be returned to “suspended” mode. You can verify that this process has worked correctly by sending mail to these users after the migration process is complete. You could also check the status of the accounts in the Directory.

Make sure to test the box migration phase on both your proxy/relayed accounts and on your non-proxy/non-related accounts to make sure that everything works correctly in both instances.

4

Troubleshooting Migration

This section deals with a variety of problems that may occur during the migration process. It also describes errors that may arise when a migration command is issued. The chapter is broken down into sections for each of the Migration Toolbox Utilities, whose function is described in Chapter 2.

The topics covered are:

- `immigacctdump` errors
- `immigacctcreate` errors
- `immigacctsort` errors
- `imboxmigrate` errors

4.1 immigacctdump Errors

The `immigacctdump` utility is used to write account data from a source system to a text file, which is used as input to populate account information in InterMail's Integrated Services Directory. (See Chapter 2 of this manual for a discussion of creating an input file using the `immigacctdump` utility.)

Errors produced during the `immigacctdump` process are generally caused by incomplete or incorrect records in the input file. Error messages are meant to notify you that certain information has not been found, or has been skipped, for example:

Error	Description and possible action
`Account has POP3 and UNIX delivery: address`	Only POP3 mailbox will be migrated. A warning that an account has two mailboxes and only one of them will be transferred to InterMail.

In this instance, an error condition is reported, but a record of some kind is still written to the dump file for this account. However, most messages indicate a complete failure to dump an account record. A few examples of these errors and the suggested courses of action (if appropriate) are as follows:

Error	Description and possible action
`No password found for username`	The password entry is not valid, so the account will be set up without one. Check for incorrectly placed or missing `':'` characters in the `passwd` file.
`Mailing list skipped: address`	Migration of mailing lists is not supported.

`Unsupported forward for username ignored: "forward"`	Pipes to programs, deliveries to files, and :include: files are not supported.

Note: *One method to streamline the* `immigacctdump` *process and speed the troubleshooting of errors is to use the* `-p` *option with* `immigacctdump`. *This causes* `immigacctdump` *to stop processing* `/etc/passwd` *file entries if it encounters an incorrectly formatted entry. The summary printed in the resulting error output from* `immigacctdump` *will list the accounts that were found, and you can compare this to the number of accounts you actually expected.*

Reported errors for Post.Office and sendmail account dumps are somewhat different. The next two sections provide complete error lists for each.

4.1.1 Post.Office Dump Errors

The following list of errors may occur during Post.Office `immigacctdump` runs.

Error	Description and Possible Action
`Failed to look up account: account_id`	The specified account could not be looked up in the account database. Verify the account.
`Skipping default account: account_id`	Default accounts are not really accounts, so they are skipped.
`Skipping remote account: account_id`	The account is for a remote subscriber to a mailing list, so it is ignored.
`Skipping manager account: account_id`	The "Manager Account" is a built-in Post.Office account Post.Office account that has no use in InterMail, therefore it is skipped.
`Mailing list skipped: address`	Mailing list migration is not supported, so this account is skipped.
`Account has POP3 and UNIX delivery: address (only POP3 mailbox will be migrated)`	A warning that an account has two mailboxes and only one of them will be transferred to InterMail.
`Account is missing a UNIX username: address (no mailbox will be migrated)`	An account is set up for UNIX delivery but doesn't have a UNIX id specified.
`Account has an invalid UNIX username (username): address (no mailbox will be migrated)`	The UNIX id specified in the account was not found in the `passwd` database.
`No UNIX mailbox found for username: address (no mailbox will be migrated)`	No mailbox was found within the list of directories to search (`-m` option).
`Account has no deliveries: address`	The account will not receive any mail since it doesn't get delivered to a mailbox or forwarded to another address.
`Failed to look up alias: address`	The alias could not be looked up to determine where it forwards.

4.1.2 Sendmail Dump Errors

The following of errors may occur during sendmail `immigacctdump` runs.

Error	Description and Possible Action
`No password found for username`	The `passwd` entry for `username` isn't valid, so the account will be set up without a password.
`No mailbox found for username`	No mailbox was found within the list of directories to search (`-m` option). No mailbox will be migrated; but as long as the user's account information is in the InterMail ISD, a mailbox will be created the next time he/she receives a message. ***Note:*** *This applies to all "no box" errors.*
`Unsupported forward for username ignored: ``forward''`	Pipes to programs, deliveries to files, and :include: files are not supported.
`Malformed alias (line #, alias)`	The alias has invalid syntax.
`Orphaned continuation line in alias file (line #)`	A continuation line was encountered without a preceding alias.
`:include: aliases are not supported (line #, alias`	The alias expansion contains an :include: file. This is not supported.
`Piping to programs is not supported (line #, alias)`	The alias expansion contains a pipe to a program. This is not supported.
`Appending to files is not supported (line #, alias)`	The alias expansion contains a filename. This is not supported.
`\username aliases are not supported (line #, alias)`	The alias expansion contains a local delivery of the form \username. This is not supported.
`Alias has no recipients: alias`	No supported deliveries were found for the alias so it wasn't written to the output.

Note: *During a dump of sendmail accounts using a custom* `passwd` *file (*`-p` *option),* `immigacctdump` *will stop processing* `passwd` *entries if an incorrectly-formatted entry is encountered. The summary printed in the error output lists the number of accounts that were found, so this number should be compared to the expected number of accounts. Typically, the problem with* `passwd` *entries is that they are missing some fields (not enough* `':'` *characters). Once the entries are fixed, the dump should proceed normally.*

4.2 immigacctsort Errors

Errors caused by `immigacctsort` are either due to badly formed syntax in the command line, a non-existent (bogus) parameter, or a non-existent filename as shown below:

Error	Description and Possible Action
`No sort parameters were specified.`	One parameter is required in this operation. Add a parameter.
`Invalid parameter: ``param''`	The parameter entered is not a recognized parameter (misspelled or bogus parameter). Check spelling and start again.
`Couldn't open filename`	Either the input or output filename could not be opened. Find correct name for `immigacctdump.out` file to sort.

4.3 immigacctcreate Errors

The `immigacctcreate` utility is used to create accounts in InterMail, and can also be used to change the status of user accounts—as, for example, from "active" to "maintenance" mode. (See Chapter 2 for a discussion of maintenance mode.)

This section discusses common `immigacctcreate` errors:

4.3.1 Environment Conditions

If an error results from an incorrectly set environment variable (e.g., `$INTERMAIL` `$ORACLE_HOME`), `immigacctcreate` will exit and terminate the process. Typically, the fix for these errors is to set the environment variable and change permissions on a file or directory. Some examples of errors and suggested courses of action are shown in the following table:

Error	Description and Possible Action
`$INTERMAIL environment variable not set, stopped at immigacctcreate line xxx`	Set `$INTERMAIL`.
`Could not open ORACLE_HOME: $oracleDir directory, stopped at immigacctcreate line xxx`	Set `$oracleDir`, add read permission.
`SQLPLUS program: $sqlplusFile is not executable, stopped at immigacctcreate line xxx`	Add execute permission to `$sqlplusFile`.
`immigacctcreate: Couldn't open $hostnameFile`	Add read permission to `$hostnameFile`.
`Warning: Directory DB connection information not found in config.db` `(Checking environment, ORACLE_USERINFO and ORACLE_DB_SERVICE)`	Add config.db keys for oracle connections.

Input Data Validation Errors

Accounts that are created successfully are written to the `immigacctcreate<pid>.pass` file. If records in the output file used by `immigacctcreate` are incorrectly formatted or are missing information, these records will fail and be written to the `immigacctcreate.<pid>.fail` file.

The reason for an `immigacctcreate` error is shown in the `AcctCreate-Error` field in the fail file. The following illustration of a fail file record shows an error caused by a `missing POP-Login, Password` or `Password-Type`.

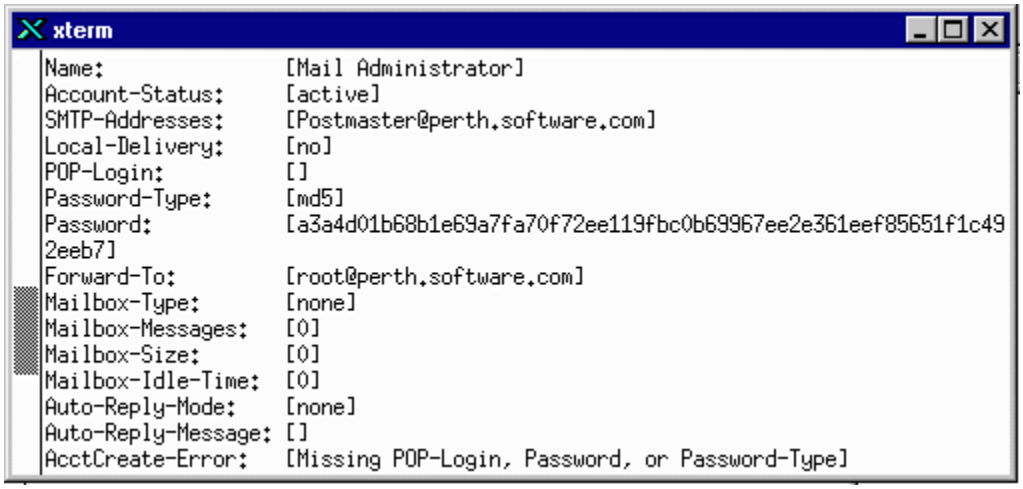

Figure 3 immigacctcreate.<pid>.fail file

The solution is to go through the fail file, record by record, fixing the indicated problems. Once this is done, you can submit the corrected fail file as input for another `immigacctcreate` run.

4.4 imboxmigrate Errors

The `imboxmigrate` utility is used to move mailboxes and their associated messages from a legacy mail system onto an InterMail MSS host. The mailbox migration process creates pass files, which are described in Chapter 2, and fail files, which are discussed here.

The following illustration gives an example of a typical `imboxmigrate.<pid>.fail` file and the cause of an account record error.

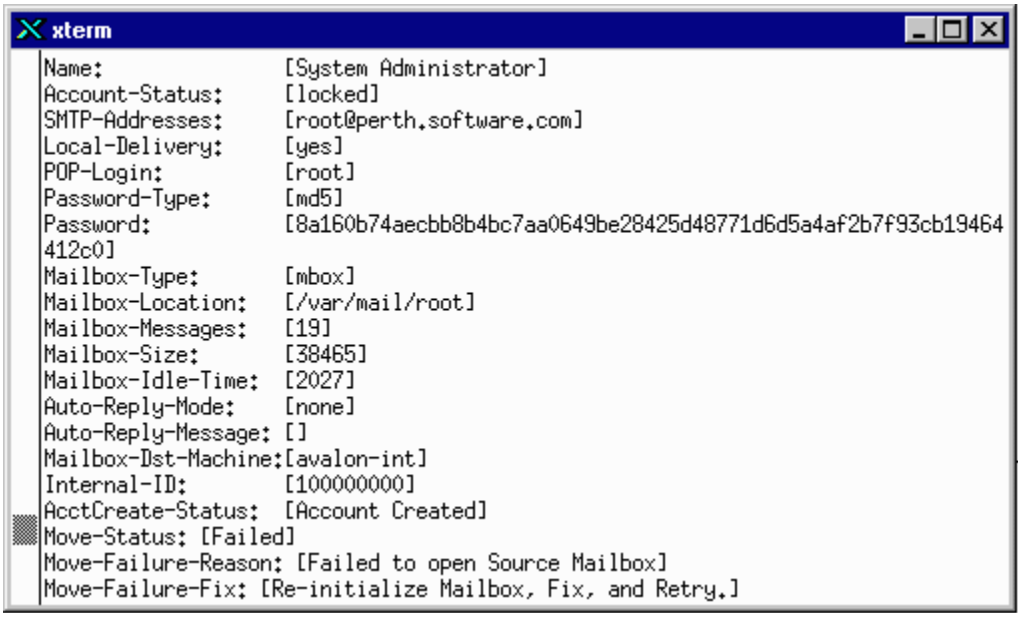

Figure 4. imboxmigrate.<pid>.fail file

The example record from an `imboxmigrate.<pid>.` fail file gives diagnostics in the `Move-Failure-Reason` field. This tag helps you determine the reason for the box migration failure, also offers a solution for fixing the problem.

4.4.1 imboxmigrate Fail Files

The `imboxmigrate` command creates a fail-file containing account records for mailboxes that were not successfully moved during migration. This file also reports if the mailbox comparison phase for that account failed. Each of the fail-file entries is appended with information that specifies the reason for failure as best as could be determined by `imboxmigrate.` The filename is:

```
imboxmigrate.<Process-id>.[ pass | fail ]
```

`imboxmigrate` sends error output to standard-error and other informational output to standard-out. Informational output includes the following:

- When the process starts, a message is printed that describes the actions the program will take.
- When finished with a mailbox (either a migration success or failure), the account name is printed out.

Some of this information is also written to the log directory in the `imboxmigrate.<date&time>.log` and `.trace` files, as well as the MSS log files.

Fail File Content

The fail-file will contain the following additional lines for each entry:

```
Move-Status: [Failed]
Move-Failure-Reason: [UNKNOWNUSER]
Move-Failure-Fix: [Create Mailbox and put in Maintenance Mode]
```

The Mailbox-Failure-Reason will be one of the following::

```
Failed to open Source Mailbox for Copy
Failed to create or open Destination MSS
Source Mailbox Type Unknown
Destination Mailbox Type Unknown
Cannot Create Folder in Destination Mailbox
Cannot Write Message in Destination Mailbox
Folder missing in Destination Mailbox
Extra Folder in Destination Mailbox
Validation of Destination type is NYI
Failed to set AutoReplyText in destination
Source file is corrupted
Did not find inbox in destination message store
Move failed
Failed to open Source Mailbox for Comparison
Failed to match all source messages in destination
Failed to open Destination Mailbox for Comparison
Cannot Read a Message in Source Mailbox
Source Mailbox could not be read for Comparison
Invalid account status for migration
```

The `imboxmigrate` command generates entries in a log file in the log directory configured for the InterMail installation. This file has the following name convention:

```
$INTERMAIL/$logDir/imboxmgrate.YYYYMMDDHHMMSSmmm.log
```

5

Toolbox Utility Syntax

This chapter contains the complete syntax of the four Migration Toolbox utilities and a brief discussion of `immigdbcontrol`. It is meant to serve chiefly as a reference for the migration tools whose use is described in Chapter 2.

The follow topics are discussed:

- `immigacctdump`
- `immigacctsort`
- `immigacctcreate`
- `imboxmigrate`
- `immigdbcontrol:`

5.1 The immigacctdump Utility

The following syntax is used for `immigacctdump`. For an operational discussion of this utility, refer to Chapter 2 of this manual.

Usage

```
immigacctdump -h
immigacctdump -tp [-c <file>] [-q <quota>] [-m <directory>]
                  [-o <file>] [-e <file>]
immigacctdump -ts -d <domain> -m <directory> [-f <directory>]
                  [-q <quota>] [-p <file> [-s]] [-a <file>]
                  [-D <domain>] [-x] [-o <file>] [-e <file>]
```

Where:

- `-h` Prints usage information.
- `-t <type>` Type of system being migrated, either `'p'` for Post.Office or `'s'` for sendmail.
- `-o <file>` File to collect output; by default, the output is written to standard output.
- `-e <file>` File to collect error output; by default, the error output is written to standard error.

Post.Office options:

- `-c <file>` Specifies a custom configuration file to use instead of the default of `/etc/post.office.conf.`
- `-q <quota>` Mailbox quota to set up for users that don't have one (specified in kilobytes).
- `-m <directory>` For users who have mail delivered to their UNIX mailboxes, look in this directory for their `mbox` files. Several locations can be specified and will be searched in the same order as they appear on the command line. The `'~'` character is special and means to look in users' home directories for a file named `mbox.`

sendmail options:

- `-d <domain>` Set up all users' addresses in this domain. Several domains can be specified to set up users in multiple domains.
- `-m <directory>` Look in this directory for users' `mbox` files. Several locations can be specified and they will be searched in the same order as they appear on the command line. The '~' character is special and means to look in users' home directories for a file named `mbox.`
- `-f <directory>` Look in this directory for users' forwarding files Several locations can be specified and they will be searched for in the same order as they appear on the command line. The '~' character is special and means to look in users' home directories for a `.forward` file Otherwise a file named after the users' logins is sought.
- `-q <quota>` Mailbox quota to set up for all users (specified in kilobytes).
- `-p <file>` Specify a custom password file to be used for migrating accounts. If unspecified, all current system accounts will be migrated. The file is expected to contain users' encrypted passwords.
- `-s` When a custom password file is specified, the encrypted passwords are assumed to be in that file; this option causes `immigacctdump` to look up missing passwords in the shadow password database.
- `-a <file>` Indicates that aliases should be dumped in addition to accounts, and the named file used as the source for alias definitions.
- `-D <domain>` The domain used to complete short hostnames in addresses; by default, the completion domain is found in `/etc/resolv.conf.`
- `-x` Only dump accounts that have a mailbox file.

5.2 The immigacctsort Utility

The following syntax is used for `immigacctsort`. For an operational discussion of this utility, refer to Chapter 2 of this manual.

Usage

```
immigacctsort -h
immigacctsort [-i <file>] [-o <file>] sort_param ....
```

`-h` prints usage information

`-i <file>` file to use as input; by default, the input is read from standard input

`-o <file>` file to collect output; by default, the output is written to standard output

`sort_param` a list of parameters to compare when sorting the input, such as Mailbox-Size; accounts that don't contain the parameter sort to the end of the output; a second or subsequent parameter is used only when all previous parameters match exactly.

The available sort parameters are:

- `Name`
- `Account-Status`
- `SMTP-Addresses`
- `Local-Delivery`
- `POP-Login`
- `Password-Type`
- `Password`
- `Forward-To`
- `Mailbox-Type`
- `Mailbox-Location`
- `Mailbox-Messages`
- `Mailbox-Size`
- `Mailbox-Idle-Time`
- `Mailbox-Quota`
- `Auto-Reply-Mode`
- `Auto-Reply-Message`

5.3 The immigacctcreate Utility

The following syntax is used for `immigacctcreate`. For an operational discussion of this utility, refer to Chapter 2 of this manual.

Usage

```
immigacctcreate -help
immigacctcreate -h <host> -a <file> [-m |-p [-x <host> ] [-r <host> ]]
[-u] [-o file] [-e file]
```

Where:

- `-help` prints this usage information
- `-h <host>` MSS host name
- `-a <file>` specifies a file containing input account data. The format matches `immigacctdump` output format.
- `-m` forces account status to "maintenance" as accounts are created or updated, regardless of the account status specified in the input data default `off`.
- `-p` forces account status to "proxy" as accounts are created or updated, regardless of the account status specified in the input data default `off`.
- `-x <host>` individual pop proxy host name. If a pop proxy host name is not specified here, the script inserts a blank value, which causes InterMail to use the global pop proxy host from config.db. This is only used with `"-p"` option.
- `-r <host>` individual SMTP relay host name If this is not specified here, the script inserts a blank value, which causes InterMail to use the global SMTP relay host from config.db This is only used with `"-p"` option.
- `-u` forces "update" rather than "create."' Accounts are updated with the input account data. An error occurs if the account does not already exist The default setting is "off." When "update" mode is specified, either:

  a.) If proxy mode is specified the new (overloaded) values are entered into MSS host (`-x host` value) and auto-reply host (`-r host` value), or

  b.) Values from the input file are entered into MSS host and auto-reply host.

- `-o <file>` files to collect output (both `STDOUT` and `STDERR`).
- `-e <file>` and progress indications. All data and intermediate output is directed `to .pass, .fail, .alias,` and `.forward` files.

5.4 The imboxmigrate Utility

The following syntax is used for `imboxmigrate`. For an operational discussion of this utility, refer to Chapter 2 of this manual.

Usage

The `imboxmigrate` command migrates the mailboxes of users from Berkeley mailbox format files or from Post.Office format directories to an InterMail Message -Store Server (MSS). The `<batch_file_name>` parameter to the command specifies a dump file that contains entries that identify the users to be moved. `imboxmigrate` works in conjunction with `immigacctdump` which creates the dump file in the right format, and with `immigacctcreate,` which creates accounts for those users in the InterMail Message Store.

The following flags can be specified in addition to the parameter that specifies the batch file:

- `-append | -force | -create | -forcewarn`

These determine the mode in which the destination message store will be accessed. `-append` will fail if the mailbox does not exist. `-force` will create a message store if it is not present. `-create` will fail if the mailbox already exists. `-forcewarn` will create a message store if it is not present, but a warning message will be logged. The default is `-create`.

Validation Phase

After the messages have been migrated, you can run a validation phase to compare the original messages with the stored messages.

- `-compare | -inbox | -nocompare`

The `-compare` option specifies that all original messages are compared with all the newly stored messages. The `-inbox` option specifies that only messages in the "inbox" will be compared. The `-nocompare` option turns off the validation phase. The default setting is `-compare`.

If the validation phase fails, `imboxmigrate` will *not* delete the copied messages or otherwise restore the destination mailbox to its initial state.

Threads

You can set the number of threads used to move mailboxes. Each thread moves one mailbox at a time. The flag for this is:

- `-threads <int>`

This specification overrides the value set using the `imboxmigrateNumThreads` configuration key. (see Chapter 11 of the *InterMail Reference Guide).* If there is no specification, the default is 30. To set a value, substitute a number for `<int>`. The maximum recommended value is 2000.

Help

The help flag provides help with command usage. No actions are taken.

- `-help`

Pass Files

The `imboxmigrate` command creates a pass-file containing the entries that were successfully moved (and compared if specified) and a fail-file containing those entries that were not successfully moved, or if the comparison phase for that account failed. Each of the fail-file entries is appended with information that specifies the reason for failure as best as could be determined by `imboxmigrate.` The filename is:

```
imboxmigrate.<Process-id>.[ pass | fail ]
```

`imboxmigrate` sends error output to standard-error and other informational output to standard-out. Informational output includes the following:

- When the process starts, a message is printed that describes the actions the program will take.
- When finished with a (either successfully or by failing to move it), the account name is printed out.

Some of this information is also written to the log directory in the `imboxmigrate.<date&time>.log` and `.trace` files.

5.5 The immigdbcontrol Utility

This section offers reference information for using `immigdbcontrol.` It is not meant as an operational guide for creating domains, modifying account information and so forth. Since this utility is functionally equivalent to `imdbcontrol`, you should refer to Chapter 3 of the *Integrated Services Directory Guide* for a complete operational discussion and examples for using this utility.

5.5.1 Syntax

The syntax for using `immigdbcontrol` is as follows:

```
immigdbcontrol [-d] [-u <user/password@db>] [-h] [-] <option>
```

Where:

`-d` Prints progress information for the operation.

`-u` Allows specification of database administrator information, in the form:

```
username/password@db-service
```

For Example:

```
imail/imail@IMD1
```


`-h`	Displays usage information for the specified option. For example, entering `immigdbcontrol -h CreateAccount` returns usage information (number and type of parameters, etc.) for the `CreateAccount` option.
`-`	Specifies that multiple lines should be processed from STDIN and executed one-by-one, just as if each line was invoked with a separate invocation of `immigdbcontrol`. (This increases the performance of `immigdbcontrol` by a factor of about 5 in doing batched operations.)
`option`	Specifies the operation to be performed (see table below).

The particular operation performed by a given `immigdbcontrol` instruction is specified by a command-line option. The available `immigdbcontrol` options are divided between operations involving domains, e-mail accounts, forwarding addresses, SMTP Alias addresses, and system logging.

The `immigdbcontrol` command line options are case-insensitive. For example, you can create an account by entering any of the following:

```
immigdbcontrol CreateAccount ...
immigdbcontrol createaccount ...
immigdbcontrol CrEaTeAcCoUnT ...
```

5.5.2 Execution Options

The following tables provide information on the execution options available with `immigdbcontrol`.

Domain Operations

`CreateDomain`	Creates a mail domain.
`UpdateDomain`	Modifies an existing domain.
`DeleteDomain`	Deletes an existing mail domain.
`ListDomains`	Shows a list of domains for which InterMail accepts mail.
`SetDefaultDomain`	Sets the default mail domain.
`GetDefaultDomain`	Gets the default mail domain.
`SetWildcardAccount`	Sets the wildcard account for a local domain.
`UnsetWildcardAccount`	Disables wildcard delivery for a domain.
`ValidateDomain`	(This option is reserved for future use.)

E-mail Account Operations

`CreateAccount`	Creates an account.
`GetAccount`	Gets data for an existing account.
`ModifyAccount`	Modifies an existing account.
`DeleteAccount`	Deletes an existing account.
`ListAccounts`	Shows a list of all accounts.
`SetPassword`	Sets the password for an existing account.
`GetPassword`	Gets the password for an existing account.
`SetAccountStatus`	Sets the status of an account (active, locked, proxy, etc.)
`SetAccountQuota`	Sets mailbox quotas for an account.
`EnablePOPDelivery`	Enables local delivery (POP and IMAP) for an InterMail account.
`DisablePOPDelivery`	Disables local delivery (POP and IMAP) for an account.
`EnableForwarding`	Enables forwarding for an account.
`DisableForwarding`	Disables forwarding for an account.
`SetServiceLevel`	Enables or disables an e-mail service for an account.
`GetServiceLevel`	Gets information on the e-mail account services for an account.
`SetAutoReply`	Sets the auto reply mode for an account.
`SetAutoReplyHost`	Sets the location of an account's auto-reply message.

Mail Forwarding Operations

`CreateLocalForward`	Creates a local forwarding address for an existing InterMail account.
`DeleteLocalForward`	Deletes an existing local forwarding address.
`CreateRemoteForward`	Creates a remote forwarding address for an existing InterMail account.
`DeleteRemoteForward`	Deletes an existing remote forwarding address.
`ListAccountForwards`	Shows a list of forwarding addresses associated with an InterMail account.

SMTP Alias Operations

`CreateAlias`	Creates an SMTP alias for an existing InterMail account.
`DeleteAlias`	Deletes an existing SMTP alias.
`ListAliases`	Shows a list of SMTP aliases.

Index