

Quest® Migrator for GroupWise 4.7  
**Program Parameters Reference**



© 2019 Quest Software Inc.

**ALL RIGHTS RESERVED.**

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.  
Attn: LEGAL Dept.  
4 Polaris Way  
Aliso Viejo, CA 92656

Refer to our website ([www.quest.com](http://www.quest.com)) for regional and international office information.


**Patents**


Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at [www.quest.com/legal](http://www.quest.com/legal).

**Trademarks**

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at [www.quest.com/legal](http://www.quest.com/legal). All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

**Legend**

 **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

 **IMPORTANT, NOTE, TIP, MOBILE, or VIDEO:** An information icon indicates supporting information.

# Contents

<b>Introduction</b> .....	<b>5</b>
How to use this reference .....	5
About INI files .....	6
How to specify or edit program parameters .....	7
<b>Parameters for Administrator Components</b> .....	<b>8</b>
addproxy.ini (AddProxy) .....	8
addproxy.ini [AddProxy] section .....	8
adobjmerge.ini (AD Object Merge Utility) .....	11
adobjmerge.ini [ActiveDirectory] section .....	11
adobjmerge.ini [ActiveDirectory2] section .....	17
adobjmerge.ini [ErrorsToIgnore] section .....	17
adobjmerge.ini [ErrorsToWarn] section .....	18
adobjmerge.ini [General] section .....	18
adobjmerge.ini [PowerShell] section .....	19
gwdirapp.ini (Directory Exporter) .....	19
gwdirapp.ini [Config] section .....	20
gwdirapp.ini [ErrorsToIgnore] section .....	22
gwdirapp.ini [ErrorsToWarn] section .....	22
gwdirapp.ini [General] section .....	23
gwdirapp.ini [GroupWise] section .....	23
gwdirapp.ini [userlist] section .....	24
gwdirapp.ini [Users] section .....	24
gwmigapp.ini (Administrator-Driven Batch Migrator) .....	24
gwmigapp.ini [ActiveDirectory] section .....	25
gwmigapp.ini [ArchiveData] section .....	29
gwmigapp.ini [ErrorsToIgnore] section .....	29
gwmigapp.ini [ErrorsToWarn] section .....	30
gwmigapp.ini [Exchange] section .....	30
gwmigapp.ini [Filter] section .....	38
gwmigapp.ini [Filters] section .....	38
gwmigapp.ini [FolderNames] section .....	38
gwmigapp.ini [General] section .....	39
gwmigapp.ini [GroupWise] section .....	42
gwmigapp.ini [Outlook] section .....	48
gwmigapp.ini [PowerShell] section .....	48
gwmigapp.ini [ServerData] section .....	49
gwmigapp.ini [Status] section .....	49
gwmigapp.ini [X400Domains] section .....	50
<b>Parameters for the SSDM</b> .....	<b>51</b>
gwdtapp.ini [ArchiveData] section .....	51
gwdtapp.ini [ErrorsToIgnore] section .....	52

gwdtapp.ini [ErrorsToWarn] section .....	52
gwdtapp.ini [Exchange] section .....	52
gwdtapp.ini [Filter] section .....	55
gwdtapp.ini [Filters] section .....	55
gwdtapp.ini [FolderNames] section .....	56
gwdtapp.ini [General] section .....	56
gwdtapp.ini [GroupWise] section .....	61
gwdtapp.ini [PAB] section .....	64
gwdtapp.ini [Screens] section .....	64
gwdtapp.ini [ServerData] section .....	65
<b>About us</b> .....	<b>67</b>
Technical support resources .....	67
<b>Index</b> .....	<b>68</b>

---

# Introduction

Computer programs perform their functions according to sets of program parameters that specify information about how the programs should behave. Program parameters specify configuration options, environmental conditions, and many other user preferences and choices. Many MFG parameters are specified by an admin in the course of working through a program's normal user interface (screen displays). Other parameters control other aspects of program behavior that are less commonly customized.

MFG programs use INI files to store program parameters. An INI file is a simple text file, saved under a filename with an .ini extension. The program associated with an INI file can write to the INI file to preserve its operational parameters for future program runs. It is more efficient for the program to simply read this information from an INI file and load it into its on-screen fields rather than ask the user to enter it again. INI files also contain parameters for many program functions that are not exposed in the user interface, simply because not many users choose to deviate from their defaults.

This document lists and explains the program parameters that control certain operations of the component applications of Quest Migrator for GroupWise.

- [How to use this reference](#)
- [About INI files](#)
- [How to specify or edit program parameters](#)

## How to use this reference

This material is organized to help you find pertinent information quickly. The parameter definitions are divided into two primary sections, one for MFG component applications that are run by a migration administrator, and the other for MFG's Self Service Desktop Migrator (SSDM) application. Within the section for administrator components, the parameters appear in separate sections by component name, with the names of the INI files to which they correspond:

addproxy.ini: AddProxy Utility  
adobjmerge.ini: AD Object Merge Utility  
gwdirapp.ini: Directory Exporter  
gwmigapp.ini: Administrator-Driven Batch Migrator

Then, within each component section, the parameters are grouped by INI-file section, in alphabetical order as shown in this example:

[ActiveDirectory]  
[ArchiveData]  
[Exchange]  
[Filter]  
[General]

This organization lets you browse the parameters by component and section, to see what controls are available for a particular aspect of a particular component. But the document also provides an Index that lets you look up a parameter definition alphabetically, by parameter name.

# About INI files

- [Parameter format and structure](#)
- [GUI-set and other “auto-set” parameters](#)
- [Transient “scratch” parameters](#)
- [Parameter defaults](#)
- [Parameter value data types](#)

## Parameter format and structure

Within an INI file, each parameter definition takes the form of a key name followed by an equals sign (“=”) and the parameter value, like these examples:

```
Provider=quest
MigrationID=sitraka
User=proxyuser@sitraka.com
Force=1
ServerMailDest=PST
SelectedPstDir=c:\Program Files\Outlook\Archive
AllowAdd=1
```

Key names are case-insensitive, and may not contain any spaces. Spaces may surround the equals sign, and in that case the spaces are ignored.

Upper/lowercase within a parameter value is significant only if case matters in whatever context the string will be inserted. For example, if *CityName=baLtlmOre* and the program will carry that parameter value into the user interface as a display element, the name will appear there in the same peculiar form, as “baLtlmOre.”

Spelling of key names and parameter values is critical. If a program is looking for a key name *UserAddress*, it will not recognize a key name *UsreAddress* or *UserAdress*.

Program parameters are grouped into named sections, identified by section names enclosed within square brackets, like this:

```
[ThisSection]
KeyName1=blue
KeyName2=654321
KeyName3=kangaroo

[ThatSection]
KeyName4=Casablanca
KeyName5=0
```

Sections are customarily separated by blank lines, as in this example, for improved readability. Section names are case-insensitive, and may or may not contain any spaces.

## GUI-set and other “auto-set” parameters

Many parameters are automatically set by an application to record entries in the graphical user interface (GUI) of an application. That is, an application will write the parameter value to the Task Parameters upon an admin’s or user’s entry in the GUI. Many other parameters are auto-set by the Setup program that installs the Quest software, or are auto-set by a program for internal controls.

**! CAUTION:** These auto-set parameters are particularly important to application performance, and changes to auto-set values can cause the associated applications to perform erratically. *Quest strongly recommends these auto-set parameters not be manually edited except at the explicit direction of Quest Support staff, or if explicitly directed in the application notes in this document.*

## Transient “scratch” parameters

Many parameters are internal transient parameters, used by programs as temporary repositories for “scratch” data to pass information to another program, or from one subroutine to another. Any manual manipulation of these parameters is meaningless and will have no effect on the program or data, since any preexisting values are overwritten by the program as soon as it runs. All transient “scratch” parameters are clearly identified as such in this document.

## Parameter defaults

The default value of a required parameter is the value that was set when the software was originally installed—before the associated program or anyone has had a chance to edit the file. The default value of an optional parameter is the value that prevails if the parameter does not appear within the INI file.

## Parameter value data types

For any given parameter, the parameter value will be meaningful only if it conforms to the parameter’s particular data type. Every parameter is one of these five data types:

- **String:** An alphanumeric string.
- **Integer:** A non-decimal numeric value.
- **Boolean:** A single-digit value, 1 or 0, to specify whether a particular feature or function is to be enabled or disabled, or whether a particular operating condition is true or false. These values are specified by a single-digit of 1 (on, yes, or true) or 0 (off, no, or false).
- **Date:** A date, formed by eight digits with two slash characters, in the form: mm/dd/yyyy.
- **Keyword:** An alphanumeric string that matches one of a finite set of valid keyword strings defined for the particular parameter.

# How to specify or edit program parameters

The INI file associated with an MFG program resides in the same path as the program. Since an INI file is a simple text file, you can use Windows Notepad or any text editor to open, view, and edit the file.

To edit an INI file:

- 1 Use Notepad or some other text editor to open the file.
- 2 Find the pertinent [SectionName] for the parameter whose value you want to specify or edit.
  - ! **CAUTION:** Some parameters in different sections share the same keyname, so it is very important that parameter values be entered and edited in the correct sections.
- 3 Add the parameter or edit its value, as appropriate.
- 4 Repeat steps 2 and 3 as often as necessary to make as many additions and changes to the parameters as needed.
- 5 Save the changed file and close the text editor.

# Parameters for Administrator Components

- [addproxy.ini \(AddProxy\)](#)
- [adobjmerge.ini \(AD Object Merge Utility\)](#)
- [gwdirapp.ini \(Directory Exporter\)](#)
- [gwmigapp.ini \(Administrator-Driven Batch Migrator\)](#)

## addproxy.ini (AddProxy)

### addproxy.ini [AddProxy] section

#### addproxy.ini [AddProxy]

Auto-set parameters — do not edit manually:

LicenseKey= MigrationID=

#### addproxy.ini [AddProxy] FailureLog=<path>

*string; default = [none]*

Specifies the name and location of a *csv* file to which failed proxy-update attempts will be logged. The parameter must be specified as a full UNC path, including the file name, that is valid and writeable from every desktop where AddProxy will run, as in this example:

```
FailureLog=\\SERVER\VOLUME\PUBLIC\ADDPROXY\addproxyerr.csv
```

If this parameter is omitted, no Failure Log will be written.

The first line of the Failure Log file contains the column headers. Each subsequent line represents a single unsuccessful proxy update attempt:

```
Date,DisplayName,SourceAddress,TargetAddress,ErrorText  
8/4/2005 7:51:44,Joe Blow,JBLOW.GWPO.GWDOM,Joe.Blow@company.com,Can't find...
```

#### addproxy.ini [AddProxy] SuccessLog=<path>

*string; default = [none]*

Specifies the name and location of a *csv* file to which successful proxy updates will be logged. The value must be specified as a full UNC path, including the file name, that is valid and writeable from every desktop where AddProxy will run, as in this example:

```
SuccessLog=\\SERVER\VOLUME\PUBLIC\ADDPROXY\addproxy.csv
```

If this parameter is omitted, no Success Log will be written.



The first line of the Success Log file contains the column headers. Subsequent lines represent successful proxy update attempts, one per line:

```
Date,DisplayName,SourceAddress,TargetAddress,Status 8/4/2005
7:56:14,Joe Blow,JBLOW.GWPO.GWDOM,Joe.Blow@company.com,OK
```

### **addproxy.ini [AddProxy] User=<SMTPAddress>**

*string; default = [none]*

Internet address of the administrative User who will be authorized by proxy to access other users' archives and/or mailboxes for the migration. Example:

```
User=proxyadmin@xyzcorp.com
```

### **addproxy.ini [AddProxy] UserGroupWiseAddr=<GWAddress>**

*string; default = [none]*

The GroupWise address of the *User* specified above, a required parameter if *AllowSearch=0* (see above). An example, with the proper address format:

```
UserGroupWiseAddr=userid.postoffice.domain
```

### **addproxy.ini [AddProxy] AllowAdd=<#>**

*keyword; default = 1*

Specifies an operating mode for the AddProxy program, telling the program how to add proxy rights. For example, the default setting would appear as:

```
AllowAdd=1
```

Valid values are integers in the range 0–3 inclusive. A setting of 1, 2 or 3 tells the program to add new proxy rights as described below. *AllowAdd=0* tells the program to modify existing proxy rights, to add the rights necessary for migration—useful when AddProxy cannot add new proxy rights for a new admin proxy because the prerequisites for adding new rights are not satisfied (see *AddProxy Prerequisites*, in chapter 2 of the *MFG Administration Guide*). The four valid *AllowAdd* parameter values are:

- 0: For each user, determine whether the designated proxy admin (defined by the *User* and *UserGroupWiseAddr* parameters below) is already defined among the user's proxies and, if so, add all necessary migration rights to the previously granted rights. (If the designated admin is not already among the user's proxies, the program will do nothing for that user.)
- 1 (default): For each user, add proxy rights for the designated proxy admin, then test the proxy setting and delete it if it seems to be incorrect.
- 2: For each user, add proxy rights for the designated proxy admin, and then test the proxy setting and log a warning (in the log file, path and filename specified by *FailureLog* parameter) if it appears to be incorrect.
- 3: For each user, add proxy rights for the designated proxy admin, and do not test the proxy setting.

### **addproxy.ini [AddProxy] AllowSearch=<#>**

*boolean; default = 1*

*AllowSearch=1* (the default) tells the program to search the system address book for addresses. Disabling such searches (*AllowSearch=0*) can be useful in some circumstances where system address book searches do not work correctly. If *AllowSearch=0*, you must also specify values for the *User* and *UserGroupWiseAddr* parameters (see below).

## **addproxy.ini [AddProxy] Force=<#>**

*boolean; default = 0*

*Force=0* tells the program to update a user's proxy settings just once (not every time AddProxy runs). *Force=1* tells the program to update a user's proxy settings every time AddProxy runs. The default is appropriate in most cases.

## **addproxy.ini [AddProxy] PiggyBack=<#>**

*boolean; default = 1*

Specifies whether the program should wait for the user to login to GroupWise or immediately prompt the user for a userid and password. For example:

```
PiggyBack=0
```

... tells the program to prompt immediately for the userid and password if the user is not already logged into GroupWise. The default *PiggyBack=1* tells the program to not prompt for userid and password, but rather to wait for the user to log into GroupWise, for a number of seconds specified by the *PiggyBackTimeout* parameter (see below).

## **addproxy.ini [AddProxy] PiggyBackRetry=<###>**

*integer; default = 15*

Specifies how long (in seconds) to wait between checks, if AddProxy is waiting for the user to log into GroupWise. Valid values are integers in the range 0–3600 inclusive. For example:

```
PiggyBackRetry=20
```

This parameter is meaningful only if *PiggyBack=1* (see above), and will be ignored if *PiggyBack=0*.

## **addproxy.ini [AddProxy] PiggyBackTimeout=<#####>**

*integer; default = 7200*

Specifies the time (in seconds) that AddProxy should wait for the user to login to GroupWise before exiting. Valid values are integers in the range 0–86400 inclusive. For example, to tell the program to wait one hour:

```
PiggyBackTimeout=3600
```

This parameter is meaningful only if *PiggyBack=1* (see above), and will be ignored if *PiggyBack=0*. The default 7200 tells the program to wait two hours (2 hrs x 60 mins/hr x 60 secs/min = 7200). *PiggyBackTimeout=0* tells the program to wait indefinitely.

## **addproxy.ini [AddProxy] Rights=<right1> <right2> <right3>**

*keyword; default = all*

Specifies the rights to grant to the administrative user. You may specify two or more rights by separating them with spaces, as in this example:

```
Rights=archive read private settings
```

Valid values are:

- *all*: all rights.
- *none*: no rights.
- *read*: readAppointments, readMailAndPhone, readNotes, readTasks.
- *readAppointments*: read access to appointments.
- *readMailAndPhone*: read access to mail messages and phone messages.
- *readNotes*: read access to reminder notes.
- *readTasks*: read access to tasks.

- *write*: writeAppointments, writeMailAndPhone, writeNotes, writeTasks.
- *writeAppointments*: write access to appointments.
- *writeMailAndPhone*: write access to mail and phone messages.
- *writeNotes*: write access to reminder notes.
- *writeTasks*: write access to tasks.
- *alarms*: access to alarms.
- *archive*: access to the archive.
- *notifs*: access to notifications.
- *private*: access to items marked as private.
- *settings*: access to Preferences, Rules, and Groups.

## adobjmerge.ini (AD Object Merge Utility)

The adobjmerge.ini has the following sections:

- [adobjmerge.ini \[ActiveDirectory\] section](#)
- [adobjmerge.ini \[ActiveDirectory2\] section](#)
- [adobjmerge.ini \[ErrorsToIgnore\] section](#)
- [adobjmerge.ini \[PowerShell\] section](#)

## adobjmerge.ini [ActiveDirectory] section

### adobjmerge.ini [ActiveDirectory]

Auto-set parameters — do not edit manually:

AltAddressAttribute=	SqlAuthMethod=
Attribute=	SqlPassword=
FindMethod=	SqlServer=
Host=	SqlUserName=
Password=	

### adobjmerge.ini [ActiveDirectory] AllowUpdateOfAttributes=<#>

*boolean; default = 1*

In the case where an existing AD contact will be merged with corresponding object data exported from the source, and any of the GroupWise object data does not match the corresponding attribute values in AD, this parameter determines whether the existing AD object will update (overwrite) its own attribute values with the corresponding attribute values from the GroupWise source. This feature applies to the *givenName*, *surname* and *displayName* attributes, and all custom directory attributes. For example:

AllowUpdateOfAttributes=0

... would prevent the update of AD attribute values with data from the source csv file. By default (*AllowUpdateOfAttributes=1*), the program will overwrite any values in AD that differ from the corresponding attribute values in the source.

## adobjmerge.ini [ActiveDirectory] AlwaysUseOldExchDn=<#>

*boolean; default = 1*

*AlwaysUseOldExchDn=1* (the default) tells the program to always use the old legacy Exchange distinguished name (DN) for mail forwarding, so that free/busy lookups will be directed to the Exchange server (rather than to GroupWise). This feature is useful when objects were synchronized into AD by Quest CMG Directory Connector or Microsoft's Connector, and you want free/busy lookups to work for existing Outlook 2003 users after GW users migrate, and you don't mind Outlook 2003's auto-complete feature presenting long, cryptic user names. To use this feature:

- Set *AlwaysUseOldExchDn=1* in the [ActiveDirectory] sections of both *gwmigapp.ini* and *adobjmerge.ini*, **and**
- Set *RemoveSrcAddresses=1* in the [ActiveDirectory] section of *gwmigapp.ini*.

The feature can be disabled by *AlwaysUseOldExchDn=0*, which is appropriate if objects were synched into AD by some method other than CMG's Directory Connector or Microsoft's Connector, or if you don't care about free/busy lookups for Outlook 2003 users, or if you will simply force all Outlook 2003 clients to refresh their caches, or if you prefer the shorter, "friendlier" names in Outlook 2003's auto-complete feature.

## adobjmerge.ini [ActiveDirectory] Attr<#>=<AttributeName>

*string; default = [none]*

Each *Attr<#>* parameter specifies a Contact attribute whose values the AD Object Merge Tool will merge into the corresponding AD security object record, when the program merges the Contact into the object and deletes the Contact. You may define one or more *Attr<#>* parameters to specify one or multiple attributes whose data you want to carry into the merged object records. For example:

```
[ActiveDirectory]
Attr0=FirstAttributeName
Attr1=SecondAttributeName
Attr2=ThirdAttributeName
```

The digit(s) appended to an *Attr<#>* key name differentiate one parameter from another. Multiple *Attr<#>* parameters need not appear in numerical order, but must in aggregate form an uninterrupted numerical sequence beginning with *Attr0*.

## adobjmerge.ini [ActiveDirectory] CmgExcludeFromDirSync=extensionAttribute<##>

*string; default = extensionAttribute15; new in ver 3.7.1*

Specifies the AD attribute that will determine when merged and mailbox-enabled objects should synchronize back into GroupWise—if you are provisioning AD by Quest CMG Directory Connector. (This parameter is ignored if you are provisioning by some other method.) The parameter value is a string consisting of *extensionAttribute* followed by one or two digits to designate the particular attribute.:

```
CmgExcludeFromDirSync=extensionAttribute15
```

The parameter defaults to *extensionAttribute15*, which is suitable as long as your organization is not already using *extensionAttribute15* for some other purpose. If you are already using *extensionAttribute15* (and are using the CMG Directory Connector), change the parameter value to some other unused AD attribute (within the range 1–14). Note that this parameter can be set only if *SetCmgExcludeAttr=1* (see notes for *SetCmgExcludeAttr=*, also in this [ActiveDirectory] section).

## adobjmerge.ini [ActiveDirectory] CreateUserNameFrom=<#>

*keyword ("1" or "2"); default = 1*

Tells the AD Object Merge tool whether to use the values from *TargetAddress* (*CreateUserNameFrom=1*) or *NDSUserName* (*CreateUserNameFrom=2*) when updating the *SAMAccountName* attribute values in AD security objects. The *SAMAccountName* is updated only if *ReplaceExistingUserName=1*; the *CreateUserNameFrom* parameter is ignored if *ReplaceExistingUserName=0*.

## adobjmerge.ini [ActiveDirectory] CustomAttr[#]=<AttributeName>

*string; default = [none]*

Each *CustomAttr[#]* parameter in *adobjmerge.ini* specifies a particular attribute in Active Directory to receive the values of a particular GroupWise custom attribute when the AD Object Merge Tool provisions objects into AD. Example:

```
[ActiveDirectory]
CustomAttr1=extensionAttribute14
CustomAttr2=Name_of_Existing_AD_Attrib
CustomAttr3=Name_of_Another_AD_Attrib
CustomAttr4=Name_of_Yet_Another_AD_Attrib
```

You may define as many as 20 custom attributes, *CustomAttr1* through *CustomAttr20*, but every attribute defined in *adobjmerge.ini* must correspond to an existing column in the *csv* data files.

Another set of *CustomAttr[#]* parameters (in *gwdirapp.ini*) tells MFG's Directory Exporter what data columns to create in the *csv* files for custom GroupWise attributes, so the Directory Exporter can then populate those columns with the correct data. In short, the *CustomAttr[#]* parameters in *gwdirapp.ini* map GroupWise custom attributes to particular columns in the *csv* data files, while the *CustomAttr[#]* parameters in *adobjmerge.ini* map those columns in the *csv* files to particular AD attributes.

## adobjmerge.ini [ActiveDirectory] EnableUsers=<#>

*boolean; default = 0; new in ver 3.5*

Determines whether users newly created by the AD Object Merge Tool (if *ProvisionUsers=1*) should be logon-enabled. By default (0), any such newly created users are *not* logon-enabled.

## adobjmerge.ini [ActiveDirectory] MailEnableAttempts=<#>

*integer; default = 3*

Determines the maximum number of times the AD Object Merge tool will attempt to mail-enable an object in AD before giving up and logging a failure error. The default is 3, but the parameter will accept any value in the range 1–1000, inclusive.

## adobjmerge.ini [ActiveDirectory] MailEnableRetryWaitSeconds=<##>

*integer; default = 30*

Determines the number of seconds the AD Object Merge tool will wait between attempts to mail-enable an object in AD. The default is 30, but the parameter accepts any value in the range 1–30000 inclusive (30000 secs = 500 mins).

## adobjmerge.ini [ActiveDirectory] MapiRecipientFlag=<#>

*boolean; default = 1*

Determines the value that MFG's AD Object Merge Tool will assign to the AD *mAPIRecipient* property if *UseMapiRecipientFlag=1* (elsewhere in this same [ActiveDirectory] section). If *UseMapiRecipientFlag=0*, the program will ignore this *MapiRecipientFlag=* parameter and leave the AD property at its <not set> default. For example:

```
MapiRecipientFlag=1
```

... (the default) tells the Merge Tool to set the AD *mAPIRecipient* property to *TRUE*, while a value of 0 tells the program to set the property to *FALSE*. **Note:** If *UseMapiRecipientFlag=1* and **Provision Users if not found** is selected, the AD Object Merge Tool will set AD's *mAPIRecipient* to *FALSE*. See the *UseMapiRecipientFlag* parameter below for more information about this feature.

**i** | **IMPORTANT:** The same two parameters must also be set to the same values in the [ActiveDirectory] section of *gwmigapp.ini*, so the Batch Migrator can correctly set the *mAPIRecipient* when that program is run to mailbox-enable users.

## adobjmerge.ini [ActiveDirectory] MergeContactAttribWins=<#>

*boolean; default = 0*

MergeContactAttribWins=1

... tells the program to overwrite AD user object data with the corresponding contact data when merging if both the contact and the AD user object fields contain single values (neither field is null, and neither contains multiple values). If either contains multiple values, the program will add any contact value(s) to the existing AD user object value(s). By default (*MergeContactAttribWins=0*), the program will not overwrite any AD user object data when both the contact and the corresponding AD user object fields contain values.

## adobjmerge.ini [ActiveDirectory] MergeFindBySMTP=<#>

*boolean; default = 0*

Determines which of two methods the AD Object Merge Tool will use to find a contact when merging: by the SMTP address (*MergeFindBySMTP=1*), or the default method (*MergeFindBySMTP=0*) of finding a contact by its GroupWise source address. The value of this parameter also sets defaults for two other related parameters in this [ActiveDirectory] section (see below):

	Default for MergeReqImportedFrom	Default for MergeReqProxyAddress
If <i>MergeFindBySMTP=1</i> :	0	0
If <i>MergeFindBySMTP=0</i> :	1	1

## adobjmerge.ini [ActiveDirectory] ReplaceExistingUserName=<#>

*boolean; default = 0*

Tells the AD Object Merge tool whether to update AD security objects' *SAMAccountName* attribute values. When this feature is enabled (*ReplaceExistingUserName=1*), the *CreateUserNameFrom* parameter (see separate listing above, also in this [ActiveDirectory] section) tells the program whether to use the *TargetAddress* (*CreateUserNameFrom=1*) or *NDSUserName* (*CreateUserNameFrom=2*) to update the *SAMAccountName*.

## adobjmerge.ini [ActiveDirectory] MergeReqImportedFrom=<#>

*boolean; default = 1*

Determines whether the program will require that a contact be imported from Quest CMG Directory Connector or Microsoft's Connector, rather than manually created. For example, the default:

MergeReqImportedfrom=1

... tells the program to verify that each contact was imported by the CMG or Microsoft Connector. If a contact was manually created, the program will skip the contact and note an error in the log. To tell the program to not require this, set the parameter to *MergeReqImportedfrom=0*.

## adobjmerge.ini [ActiveDirectory] MergeReqProxyAddress=<#>

*boolean; default = 1*

Determines whether the program will require that each contact contain two GroupWise addresses. For example, the default:

MergeReqProxyAddress=1

... tells the program to require that each contact contain two GroupWise addresses, and to skip a contact and note an error in the log if the contact does not contain two GroupWise addresses. (Quest CMG Directory Connector and Microsoft Connector ordinarily associate two GroupWise addresses with each contact.) Conversely, *MergeReqProxyAddress=0* tells the program to not require two addresses per contact.

## **adobjmerge.ini [ActiveDirectory] MergeX500=<##>**

*boolean; default = 0; new in ver 4.3*

Tells the AD Object Merge tool whether to copy X500 addresses into their corresponding AD user objects. For example *MergeX500=1* tells the program to merge X500 addresses into the AD user objects. By default (*MergeX500=0*) the program will not merge X500 addresses.

## **adobjmerge.ini [ActiveDirectory] ProvisionUsers=<#>**

*boolean; default = 0; new in ver 3.5*

Determines whether the AD Object Merge Tool will create a new mail-enabled security object in AD when none is found by the program to correspond with a user in the *UsersToMerge.csv* file. Such new objects will be created in the container specified by the *UserContainer* parameter, also in this [ActiveDirectory] section. By default (0), the program will not create such objects.

## **adobjmerge.ini [ActiveDirectory] PSRetryAttempts=**

*integer; default = 3; new in ver 4.3*

Determines the maximum number of times the program will retry data transmissions to Active Directory due to errors in prior attempts. In some environments, transmissions to AD can occasionally be interrupted, which may lead to incomplete provisioning. This *PSRetryAttempts* parameter is used in conjunction with *PSRetryWait* (see below) to control retries of transmissions to AD. For example, the default settings:

```
PSRetryAttempts=3
PSRetryWait=15
```

... tell the program to retry an AD transmission to a maximum of three attempts (in addition to the initial attempt), at 15-second intervals.

If the error persists through all retry attempts, the program will note the error in the log, skip the current message property or element, and move on to process the next item. The retry attempts may appear in the program logs with no other documented error or warning.

Note that these *PSRetry* parameters can also be set in *gwmigapp.ini*.

## **adobjmerge.ini [ActiveDirectory] PSRetryWait=**

*integer; default = 15 or 40 (see below); new in ver 4.3*

Sets the interval (in seconds) between retry attempts when the program will retry data transmissions to Active Directory due to errors in prior attempts. This *PSRetryWait* parameter is used in conjunction with *PSRetryAttempts* to control retries of transmissions to AD, as described above for the *PSRetryAttempts* parameter. The default is 15 seconds for migration to a proprietary Exchange, or 40 seconds for migration to Office 365.

## **adobjmerge.ini [ActiveDirectory] SetCmgExcludeAttr=<#>**

*boolean; default = 1*

Determines whether the AD Object Merge tool will populate the extension attribute designated by the *CmgExcludeFromDirSync* parameter (also in this [Active Directory] section), which in turn specifies the AD attribute that will determine when merged and mailbox-enabled objects should synchronize back into GroupWise. This *SetCmgExcludeAttr* parameter is set to 1 (on) by default, but you can set *SetCmgExcludeAttr=0* to tell the program to not set (to ignore) the *CmgExcludeFromDirSync=* parameter.

## **adobjmerge.ini [ActiveDirectory] SetRecipPolicy=<#>**

*boolean; default = 1*

Determines how the program will set the Exchange option to *Automatically update e-mail addresses based on e-mail address policy*. (The *Automatically update...* feature is an Exchange feature, offered as a check box in the Exchange Management Console, on the *E-Mail Addresses* tab of the *Data Properties* for each



mailbox.) This MFG *SetRecipPolicy* parameter specifies whether MFG should automatically enable or disable that Exchange feature for the AD objects in a designated user group. For example:

```
[ActiveDirectory]
SetRecipPolicy=1
```

... tells MFG to enable the Exchange auto-update feature (mark the Exchange check box) for the AD objects after processing. If *SetRecipPolicy=0*, the program disables the Exchange auto-updating feature (unmarks the Exchange check box). The default is 1 for any target type, including Office 365. **Note:** When mailbox-enabling, make sure that this setting for *SetRecipPolicy=<#>* matches the setting in *gwmigapp.ini* for [Exchange] *EmailAddressPolicyEnabled=<#>*.

### adobjmerge.ini [ActiveDirectory] SetUserAccountControl=<#>

*boolean; default = 0*

Tells the program whether to set the *userAccountControl* attribute value in Active Directory. If:

```
SetUserAccountControl=1
```

... then the program will set the *userAccountControl* attribute value to the integer value specified by the *UserAccountControl=* parameter (also in this [ActiveDirectory] section). By default (*SetUserAccountControl=0*), the *UserAccountControl* parameter is ignored.

### adobjmerge.ini [ActiveDirectory] SqlConnect=<alphanum>

*string; default = [none]*

A command string to effect a connection to the SQL server by parameters that will override the values entered into the GUI for *SqlServer*, *SqlAuthMethod* and *SqlUserName*. Two examples:

- *SqlConnect=Provider=sqloledb;Data Source=' <server> ';Initial Catalog=NDSMigrator;UserId='??';Password='??'*
- *SqlConnect=Provider=sqloledb;Data Source=' <server> ';Initial Catalog=NDSMigrator;Integrated Security=SSPI*

### adobjmerge.ini [ActiveDirectory] SyncObjRequired=<#>

*boolean; default = 0*

Determines whether the program must find contacts—as was required prior to the 3.1 release when it was revised to accommodate the Migrator for NDS. For each user listed in the *UsersToMerge.csv* file, if *SyncObjRequired=0* (default) and no corresponding contact is found, the program simply merges the *SourceAddress* from *UsersToMerge.csv* into the AD security object, to mail-enable the object. But if *SyncObjRequired=1*, then the program will mail-enable an object only if it finds a corresponding contact.

### adobjmerge.ini [ActiveDirectory] UseMapiRecipientFlag=<#>

*boolean; default = 1*

Determines whether MFG's AD Object Merge Tool will set the AD *mapiRecipient* property according to the *MapiRecipientFlag* parameter value (also in this [ActiveDirectory] section). If *UseMapiRecipientFlag=0*, the program will leave the AD property at its *<not set>* default. But if:

```
UseMapiRecipientFlag=1
```

... (the default), the program will set the AD property according to the *MapiRecipientFlag* parameter value. **Note:** If *UseMapiRecipientFlag=1* and **Provision Users if not found** is selected, the AD Object Merge Tool will set AD's *mapiRecipient* to *FALSE*. See the *MapiRecipientFlag* parameter above for more information about this feature.

**i** **IMPORTANT:** The same two parameters must also be set to the same values in the [ActiveDirectory] section of the *gwmigapp.ini* file, so that the Admin-Driven Batch Migrator can correctly set the *mapiRecipient* property when that program is run to mailbox-enable users.



## adobjmerge.ini [ActiveDirectory] UserAccountControl=<###>

*integer; default = 512*

Assigns the integer value to the *userAccountControl* attribute value in Active Directory, but only if this feature is enabled by *SetUserAccountControl=1*, also in this [ActiveDirectory] section. If *SetUserAccountControl=1* and this *UserAccountControl* parameter is unspecified, its value defaults to 512 (*NORMAL\_ACCOUNT*). If *SetUserAccountControl=0*, this *UserAccountControl* parameter is ignored.

## adobjmerge.ini [ActiveDirectory] UserContainer=<RDN>

*keyword; default = [none]; new in ver 3.5*

Specifies the relative domain name (RDN) of the container where the AD Object Merge Tool will create a new security object when the program finds none to correspond with a user in the *UsersToMerge.csv* file, and if *ProvisionUsers=1* (also in this [ActiveDirectory] section). For example:

```
UserContainer=OU=Users
```

... will put all new users in the *Users* container. The designated container must already exist. This *UserContainer* parameter is required if *ProvisionUsers=1*.

# adobjmerge.ini [ActiveDirectory2] section

## adobjmerge.ini [ActiveDirectory2]

Auto-set parameters — do not edit manually:

```
Host=           Password=           Userid=
```

# adobjmerge.ini [ErrorsToIgnore] section

## adobjmerge.ini [ErrorsToIgnore] Error[#]=<[#####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should ignore when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToIgnore]
Error0=[1465-32-4-00000000]
Error1=[7442-1-155-80072035]
Error2=[1400-31-108-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## adobjmerge.ini [ErrorsToWarn] section

### adobjmerge.ini [ErrorsToWarn] Error[#]=<[#####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should change to a warning when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToWarn]
Error1=[1465-32-4-00000000]
Error2=[7442-1-155-80072035]
Error3=[113-1-97-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## adobjmerge.ini [General] section

### adobjmerge.ini [General]

Auto-set parameters — do not edit manually:

EndTime=	MigrationID=	UserDoneCnt=
ErrorCnt=	Provider=	UserFailCnt=
KeyError=	RunName=	UserListHasSearchKey=
LicenseKey=	StartTime=	
LogDir=	Status=	

### adobjmerge.ini [General] EnableBPOS=<#>

*boolean; default = 0*

*EnableBPOS=1* enables the BPOS features in the AD Object Merge tool, and adds *BPOS-D* to the drop-down list of eligible migration targets. *EnableBPOS=1* also tells the AD Object Merge Tool to display the *Specify BPOS-D Identification Attributes* screen, unless *ShowBPOSIDAttr=0* (as explained for *ShowBPOSIDAttr=<#>* below).

### adobjmerge.ini [General] LogLevel=<#>

*keyword; default = 1*

Specifies how much detail the program will include in its run logs. Valid values are single-digit integers in the range 0–3 inclusive. The default *LogLevel=1* is suitable for most routine production runs. If you are having trouble and want to diagnose a problem, set the value to *LogLevel=3* (verbose).

### adobjmerge.ini [General] ShowBPOSIDAttr=<#>

*boolean; default = 1*

Determines whether the AD Object Merge tool will display the *Specify BPOS-D Identification Attributes* screen when *EnableBPOS=1* (see above). This parameter permits selection of BPOS-D as the migration target but without showing the *Specify BPOS-D Identification Attributes* screen (*ShowBPOSIDAttr=0*), in

which case all settings on that screen remain at their defaults. Note that this parameter is ignored (irrelevant) if *EnableBPOS=0*.

### **adobjmerge.ini [General] UserList=<path>**

*string; default = [none]*

Tells the program where it will find the user-list csv file, generated by the Directory Exporter, that specifies the users whose accounts the AD Object Merge Tool will merge in its current or next run. The <path> must be specified as a full UNC path including filename. For example:

UserList=C:\Program Files\Quest\Migrator for GroupWise\user\UsersToMerge.csv

## **adobjmerge.ini [PowerShell] section**

### **adobjmerge.ini [PowerShell] IdleConnectionTimeoutSeconds=<##>**

*integer; default = 30*

Specifies how long (in seconds) MFG will hold open an idle remote PowerShell connection before closing it. The default *IdleConnectionTimeoutSeconds=30* is suitable for most environments. Note that this feature applies only to the duration of an idle state during a connection. Each command execution resets this timer to zero, so a series of commands with only short idle periods between commands could keep the connection open indefinitely. *IdleConnectionTimeoutSeconds=0* tells MFG to not wait (wait 0 seconds) for a second command after a first, so every PowerShell connection would close immediately after only one command.

Remember that this parameter must be set on each migration machine, and should also be set in *gwmigapp.ini*.

### **adobjmerge.ini [PowerShell] MaxPowerShellConnections=<#>**

*integer; default = 0*

Specifies a migration-server-wide limit for the number of concurrent PowerShell connections MFG can open. This parameter is intended to eliminate the possibility of MFG exceeding Microsoft's limit of 9 simultaneous runspace connections (remote PowerShell). The recommended value is derived by:

$$(R - 1) / S$$

... where *R* is the number of simultaneous runspaces allowed by your tenant (typically 9), and *S* is the number of migration servers you will use. For example, if your limit is 9 runspaces and you intend to use one migration server, then *MaxPowerShellConnections=8 = (9-1)/1*. Or for a 9-runspace limit with 2 migration machines: *MaxPowerShellConnections=4 = (9-1)/2*. The default *MaxPowerShellConnections=0* is interpreted as "no limit," effectively turning off this limiting feature.

This parameter must be set on each migration machine, and should also be set in *gwmigapp.ini*.

## **gwdirapp.ini (Directory Exporter)**

The *gwdirapp.ini* has the following sections:

- [gwdirapp.ini \[Config\] section](#)
- [gwdirapp.ini \[ErrorsToIgnore\] section](#)
- [gwdirapp.ini \[General\] section](#)
- [gwdirapp.ini \[GroupWise\] section](#)
- [gwdirapp.ini \[userlist\] section](#)

- [gwdirapp.ini \[Users\] section](#)

## gwdirapp.ini [Config] section

### gwdirapp.ini [Config]

Auto-set parameters — do not edit manually:

LicenseKey=	Provider=	GroupsToProvisionPath=
MigrationID=	SMTPTranslationTable=	KeyError=

### gwdirapp.ini [Config] MigrateUsersSample=<path>

*string; default = [none]*

Tells the program where to save the user-list csv file that will specify to the batch-migration program which user accounts to process in its current or next run. The <path> must be a full UNC path including filename. For example:

MigrateUsersSample=C:\Program Files\Quest\Migrator for GroupWise\user\UsersToMigrate.csv

### gwdirapp.ini [Config] UsersToMergePath=<path>

*string; default = [none]*

Tells the program where to save the user-list csv file that will specify to the AD Object Merge Tool which user accounts to merge in its current or next run. The <path> must be a full UNC path including filename. For example:

UsersToMergePath=C:\Program Files\Quest\Migrator for GroupWise\user\UsersToMerge.csv

### gwdirapp.ini [Config] dumpaliases=<#>

*boolean; default = 0*

*dumpaliases=1* tells the Directory Exporter to extract users' aliases from GroupWise and insert them into the *AddressTranslation.csv* file, in the *TargetAliases* column. (While aliases are exported from GroupWise, nicknames are not. Alias and nickname are not the same data element. To export nicknames as well, see [gwdirapp.ini \[Config\] dumpNicknameAliases=<#>](#) on page 20) Then, when the Admin-Driven Batch Migrator mailbox-enables users' Exchange accounts, it can load the aliases into Active Directory. This feature is disabled (0) by default.

### gwdirapp.ini [Config] dumpNicknameAliases=<#>

*boolean; default = 0*

When *dumpNicknameAliases* is set to 1, Directory Exporter extracts users' nicknames from GroupWise, and insert them to the *TargetAliases* column in the *AddressTranslation.csv* file. When the Admin-Driven Batch Migrator mailbox-enables users' Exchange accounts, it can load the nicknames into Active Directory. This feature is disabled by default.

**Note:** To export nicknames, set both *dumpaliases* and *dumpNicknameAliases* to 1. That is, you will export the aliases as well when exporting the nicknames.

### gwdirapp.ini [Config] TargetAliasFormatTypes =<#>

*integer; default = 255*

Determines which formats of GroupWise alias will be created for the *TargetAliases* column in the *AddressTranslation.csv* file.

**Note:** This parameter only works when the parameter *dumpaliases* is set to 1.

The available alias formats and their values are described as follows:

Alias Format	Value
userid@domain	Mandatory
userid.postoffice.domain@domain	1
firstname@domain	2
lastname@domain	4
firstname.lastname@domain	8
lastname.firstname@domain	16
userid.postoffice@domain	32
domain.postoffice.userid@domain	64
postoffice.userid@domain	128

The value of the parameter *TargetAliasFormatTypes* should be the sum of all values of the alias formats you select. For example, if set to 6, the formats *firstname@domain*, *lastname@domain* and the mandatory one *userid@domain* are selected (as long as the first name and last name fields have been filled in GroupWise).

### gwdirapp.ini [Config] multipleSMTPDomain =<#>

*boolean; default = 0*

When *multipleSMTPDomain* is set to 1, users are allowed to enter their Primary Internet Domain for the *TargetAddress* attribute, and multiple Secondary Internet Domains (separated by comma) for the *TargetAlias* attribute.

**Note:** This parameter only works when the parameter *dumpaliases* is set to 1.

### gwdirapp.ini [Config] GroupType=<XXX>

*keyword; default = GDL*

Determines which GroupType value the Directory Exporter will assign to all groups in its *GroupsToProvision.txt* file. The Admin-Driven Batch Migrator in turn reads the GroupType values in *GroupsToProvision.txt* to determine the types of groups to provision in Active Directory. The value is a three-letter string defined by the combination of two elements:

<u>1st Element (one letter):</u>	<u>2nd Element (two letters):</u>
G = Global	DL = Distribution List
L = Local [domain]	SG = Security Group
U = Universal	

For example, the default setting *GroupType=GDL* signifies a Global Distribution List. Similarly, *GroupType=LSG* tells the program to set all GroupType values in *GroupsToProvision.txt* to LSG (Local Security Group).

After the Directory Exporter runs, you can edit *GroupsToProvision.txt* to change the GroupType setting for any one or more groups individually, and thereby provision groups of different types within a single run of the Admin-Driven Batch Migrator. But the Directory Exporter will initially set all of the GroupType values to whatever value is defined by this INI parameter.

### gwdirapp.ini [Config] UseAdminAPI=<#>

*boolean; default = 1*

Determines whether the program will use or ignore the Admin API. Ordinarily this is on (*UseAdminAPI=1*) by default, but *UseAdminAPI=0* tells the program to not use the Admin API — for example, if you are not using NDS DN and suspect the Admin API may be causing errors.

## gwdirapp.ini [Config] WorkerLimit=<####>

*integer; default = 1000*

Defines the maximum number of addresses per processed batch in system address book conversions. For example: *WorkerLimit=888*

This parameter can be useful if you suspect memory leaks.

## gwdirapp.ini [ErrorsTolgnore] section

### gwdirapp.ini [ErrorsTolgnore] Error[#]=<[####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should ignore when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsTolgnore]
Error0=[1465-32-4-00000000]
Error1=[7442-1-155-80072035]
Error2=[1400-31-108-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## gwdirapp.ini [ErrorsToWarn] section

### gwdirapp.ini [ErrorsToWarn] Error[#]=<[####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should change to a warning when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToWarn]
Error1=[1465-32-4-00000000]
Error2=[7442-1-155-80072035]
Error3=[113-1-97-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## gwdirapp.ini [General] section

### gwdirapp.ini [General]

Auto-set parameter — do not edit manually:

Cancel=

### gwdirapp.ini [General] FilterUser[#]=<User>

*string; default = [none]*

Tells the Directory Exporter to skip directory data for the designated <User>, used to bypass a user if his/her data generates an error in the export process, as reported in the Error Log Report. For the <User> value, enter the exact identifying text string as it appears in the *Error Log Report* (a display name value). You can enter multiple parameters to skip multiple <User>s—that is, enter a *FilterUser0* parameter for the first user you want to skip, then *FilterUser1* for the second, *FilterUser2* for the third, and so forth. Example:

```
FilterUser0=Smith, John
FilterUser1=Doe, Jane
FilterUser2=Queue, Susie
```

### gwdirapp.ini [General] LogLevel=<#>

*keyword; default = 1*

Specifies how much detail the program will include in its run logs. Valid values are single-digit integers in the range 0–3 inclusive. The default *LogLevel=1* is suitable for most routine production runs. If you are having trouble and want to diagnose a problem, set the value to *LogLevel=3* (verbose).

## gwdirapp.ini [GroupWise] section

### gwdirapp.ini [GroupWise]

Auto-set parameters — do not edit manually:

GWIALDAPPort=            GWIALDAPSearchBase=            GWIALDAPsrvr=

### gwdirapp.ini [GroupWise] CustomAttr[#]=<AttributeName>

*string; default = [none]*

Each *CustomAttr[#]* parameter in *gwdirapp.ini* specifies a GroupWise custom directory attribute whose values you want the Directory Exporter to capture and export to the *csv* data files. Another set of parameters in *adobjmerge.ini* can then map the same attributes from the *csv* file to Active Directory, so the AD Object Merge tool can migrate the attribute values when provisioning objects to AD. The *gwdirapp.ini* *CustomAttr[#]* parameter(s) here define which GroupWise custom attributes you want the Directory Exporter to capture. For example:

```
[GroupWise]
CustomAttr1=Name_of_a_GW_Attrib
CustomAttr2=Name_of_Another_GW_Attrib
CustomAttr3=Name_of_Some_Other_GW_Attrib
CustomAttr4=Name_of_Yet_Another_GW_Attrib
```

You may define as many as 20 additional attributes, *CustomAttr1* through *CustomAttr20*. The Directory Exporter then exports the values associated with the defined attributes and saves them in its *csv* data files. In short, the *CustomAttr[#]* parameters in *gwdirapp.ini* map GroupWise custom attributes to particular columns in the *csv* data files, and the *CustomAttr[#]* parameters in *adobjmerge.ini* map those columns in the *csv* files to particular AD attributes.

## gwdirapp.ini [GroupWise] GWSoapUrl=<AlphanumericString>

*string; default = [none]*

This parameter corresponds to a GUI element in the Directory Exporter: the **GroupWise Soap Server URL** field in the *GroupWise System Information* screen. The value specifies the location of the GroupWise SOAP web service. This parameter is optional, but will help resolve a duplicate-meetings issue that can occur in some scenarios in some environments where Quest CMG coexistence product is also deployed. When a GroupWise user creates a meeting in GroupWise, and then changes it (or changes one instance of a recurring series) after he/she is migrated, the invitees' calendars may show duplicate instances unless the SOAP service is enabled and its URL is specified. This solution works only for GroupWise 8 and later (not for GroupWise 7).

MFG writes this value to the INI files from an admin's GUI entry; Quest recommends you not manually enter or change this value except at the direction of Quest Support.

## gwdirapp.ini [GroupWise] InternetDomain=<domain.com>

*string; default = [none]*

Specifies the domain name the program will use to generate users' SMTP addresses, in the event the program cannot determine SMTP addresses from the information in GroupWise. (In this case, the program will append the domain name specified here to each user's GroupWise User ID and the @ symbol: *GWUserID@domain.com*.) The composed SMTP addresses are added to the *AddressTranslation.csv* file, in the *TargetAddress* field.

# gwdirapp.ini [userlist] section

## gwdirapp.ini [userlist]

All parameters in this [userlist] section are temporary "scratch" values entered and used by the program. Do not manually edit any of these values except at the direction of Quest support staff.

# gwdirapp.ini [Users] section

## gwdirapp.ini [Users]

Auto-set parameters — do not edit manually:

BadSlotSkipped=	ObjectsProcessed=	Total=
BadTypeSkipped=	OtherExported=	UnknownsSkipped=
CompaniesSkipped=	ResourcesExported=	UsersExported=
ListsExported=	Status=	

# gwmigapp.ini (Administrator-Driven Batch Migrator)

The gwmigapp.ini has the following sections:

- [gwmigapp.ini \[ActiveDirectory\] section](#)
- [gwmigapp.ini \[ArchiveData\] section](#)
- [gwmigapp.ini \[ErrorsToIgnore\] section](#)



- [gwmigapp.ini \[Exchange\] section](#)
- [gwmigapp.ini \[Filter\] section](#)
- [gwmigapp.ini \[Filters\] section](#)
- [gwmigapp.ini \[FolderNames\] section](#)
- [gwmigapp.ini \[General\] section](#)
- [gwmigapp.ini \[GroupWise\] section](#)
- [gwmigapp.ini \[Outlook\] section](#)
- [gwmigapp.ini \[PowerShell\] section](#)
- [gwmigapp.ini \[ServerData\] section](#)
- [gwmigapp.ini \[Status\] section](#)
- [gwmigapp.ini \[X400Domains\] section](#)

## gwmigapp.ini [ActiveDirectory] section

### gwmigapp.ini [ActiveDirectory]

Auto-set parameters — do not edit manually:

Domain=	HomeServer=	Password=
GroupsRDN=	Host=	Userid=
HomeMailboxStore=	LoadMethod=	

### gwmigapp.ini [ActiveDirectory] AlwaysUseOldAlias=<#>

*boolean; default = 0*

Determines whether the Admin-Driven Batch Migrator will keep the *Alias/Mailnickname* stamped onto an object in AD, rather than replacing it with a value from GroupWise, when mailbox-enabling:

```
[ActiveDirectory]
AlwaysUseOldAlias=1
```

This example tells the program to keep the value set in AD. By default (*AlwaysUseOldAlias=0*) the program will let the MFG value overwrite the existing AD value.

### gwmigapp.ini [ActiveDirectory] AlwaysUseOldExchDn=<#>

*boolean; default = 1*

Determines whether MFG will always use the old legacy Exchange distinguished name (DN) for mail forwarding, which directs free/busy lookups to the Exchange server (rather than to GroupWise). This feature is useful when objects were synched into AD by Quest CMG Directory Connector or Microsoft's Connector, and you want free/busy lookups to work for existing Outlook 2003 users after the GW users migrate, and you don't mind Outlook 2003's auto-complete feature presenting long, cryptic user names. To use this feature:

- This *AlwaysUseOldExchDn* parameter must be set to 1 in the [ActiveDirectory] sections of both *gwmigapp.ini* and *adobjmerge.ini*, **and**
- The *RemoveSrcAddresses* parameter (in *gwmigapp.ini* [ActiveDirectory]) must be set to 1.

The feature is on (1) by default, but can be disabled by *AlwaysUseOldExchDn=0* — which is the appropriate choice if objects were synched into AD by some method other than CMG's Directory Connector or Microsoft's Connector, or if you don't care about free/ busy lookups for Outlook 2003 users, or if you will simply force all Outlook 2003 clients to refresh their caches, or if you prefer the shorter "friendlier" names in Outlook 2003's auto-complete feature.

## **gwmigapp.ini [ActiveDirectory] AttrFromContact=<#>**

*boolean; default = 1*

When the Admin-Driven Batch Migrator creates an AD mailbox from an existing contact (when *MboxFromContact=1*), this parameter determines whether the contact's attribute values (phone numbers, fax numbers, departments, titles, etc.) will be preserved in the new mailbox-enabled object (*AttrFromContact=1*, the default), or will be replaced by the attribute values in MFG's csv file, from the GroupWise source (*AttrFromContact=0*).

## **gwmigapp.ini [ActiveDirectory] CanAddGroupMembers=<#>**

*boolean; default = 0*

Tells the program whether to define a new AD contact for any group member not found in AD, and add the contact as a new member to the corresponding AD group, in the container specified by the *GroupContactsContainer* parameter (also in this [ActiveDirectory] section, and mandatory if *CanAddGroupMembers=1*). By default (0), the program does *not* define new contacts.

## **gwmigapp.ini [ActiveDirectory] CanDeleteGroupMembers=<#>**

*boolean; default = 0*

Tells the program whether to synchronize any previously provisioned group in AD to match the current state of the corresponding group in GroupWise. This feature is disabled by default (*CanDeleteGroupMembers=0*).

## **gwmigapp.ini [ActiveDirectory] CmgExcludeFromDirSync=extensionAttribute<##>**

*string; default = extensionAttribute15; new in ver 3.7.1*

Specifies the AD attribute that will determine when merged and mailbox-enabled objects should synchronize back into GroupWise—if you are provisioning AD by Quest CMG Directory Connector. (The parameter is ignored if you will provision by some other method.) The parameter value is a string consisting of *extensionAttribute* followed by one or two digits to designate the particular attribute. For example: *CmgExcludeFromDirSync=extensionAttribute15*

The parameter defaults to *extensionAttribute15*, which is suitable as long as your organization is not already using *extensionAttribute15* for some other purpose. If you are already using *extensionAttribute15* (and are using the CMG Directory Connector), change the parameter value to some other unused AD attribute (within the range 1–14).

This parameter can be set only if *SetCmgExcludeAttr=1* (see notes for *SetCmgExcludeAttr*, also in this [ActiveDirectory] section).

## **gwmigapp.ini [ActiveDirectory] CreateMailboxDelay=<#####>**

*integer; default = 0*

Controls the time interval between two subfunctions of the Admin-Driven Batch Migrator process for mailbox-enabling, which might otherwise generate incorrect aliases where Active Directory accounts are replicated among multiple servers. The parameter value defines a delay in seconds, in a valid range from 0 to 30000 (inclusive), and defaults to 0.

The program disables mail for an account before it creates the mailbox, but the mail-enabled account can appear to still exist for a few seconds on any servers where the account is replicated. When the program sees a not-yet-disabled account on the other server, it mistakenly treats the new mailbox as a duplicate, appending the digit "1" to the alias to differentiate it from the other account. If your AD accounts are replicated on multiple servers, and you experience this "Alias1" problem when enabling mailboxes, use the *CreateMailboxDelay* parameter to force a delay between the mail-disabling function and the mailbox creation, to allow enough time for other servers to catch up and disable their replicated accounts. For example:

[ActiveDirectory]  
CreateMailboxDelay=15

... tells the program to wait 15 seconds after mail-disabling an account before creating the associated mailbox. Quest in-house testing suggests the resolution of this problem may require a delay as long as 33 seconds, but often much shorter. Experiment with shorter values to determine the delay required in your own environment.

### **gwmigapp.ini [ActiveDirectory] ForceAddress=<#>**

*boolean; default = 1*

If the program is mailbox-enabling and finds the primary SMTP address of an Active Directory entry to be different from the expected SMTP address, this *ForceAddress* parameter determines whether the program should change the AD entry to the primary SMTP address specified in the *targetAddress* column of the exported object data csv files. For example, *ForceAddress=0* tells the program to not change the AD address entry to the expected SMTP address. The feature is enabled (*ForceAddress=1*) by default.

### **gwmigapp.ini [ActiveDirectory] GroupContactsContainer=<ContainerRDN>**

*string; default = [none]*

Specifies a relative domain name (relative to [ActiveDirectory] *GroupsRDN*, which in turn is relative to the default naming context of the domain) to identify the sub OU in AD where new contacts will be created if *CanAddGroupMembers=1*. For example: *GroupContactsContainer=OU=Groups*

### **gwmigapp.ini [ActiveDirectory] MapiRecipientFlag=<#>**

*boolean; default = 1*

Determines the value the Admin-Driven Batch Migrator will assign to the AD *MapiRecipient* property, when mailbox-enabling, if *UseMapiRecipientFlag=1* (elsewhere in this same [ActiveDirectory] section). If *UseMapiRecipientFlag=0*, the program will ignore this *MapiRecipientFlag* parameter and leave the AD property at its <not set> default. For example, *MapiRecipientFlag=1* (the default) tells the program to set the AD *MapiRecipient* property to TRUE, while a value of 0 tells the program to set the property to FALSE. See the *UseMapiRecipientFlag* parameter below for more information about this feature.

**i | IMPORTANT:** The same two parameters must also be set to the same values in the [ActiveDirectory] section of the *adobjmerge.ini* file, so that the Batch Migrator can correctly set the *MapiRecipient* property to match the value set by the AD Object Merge Tool.

### **gwmigapp.ini [ActiveDirectory] MBoxFromContact=<#>**

*boolean; default = 0*

Tells the Admin-Driven Batch Migrator whether to convert a contact into a mailbox when mailbox enabling. This feature is off (0) by default, since in a typical scenario no contacts should remain after the AD Object Merge Tool has run (to merge contacts into existing AD objects). This feature would therefore be irrelevant, but in the event any contacts do remain most admins would prefer that the program generate an error message rather than simply convert the contact to a mailbox, which would likely yield two objects for a single user. The feature can be enabled by *MBoxFromContact=1* for some rare circumstances where that option is preferable.

### **gwmigapp.ini [ActiveDirectory] MergeGrpMembership=<#>**

*boolean; default = 1*

When merging, the default setting (*MergeGrpMembership=1*) tells the Admin-Driven Batch Migrator to add a user to a group if the corresponding Contact is a member of the group. The alternate setting (*MergeGrpMembership=0*) disables this feature.

## gwmigapp.ini [ActiveDirectory] PSRetryAttempts=

*integer; default = 3; new in ver 4.3*

Determines the maximum number of times the program will retry data transmissions to the Exchange environment due to errors in prior attempts. In some environments, lower signal quality can occasionally interrupt transmissions to Exchange. This *PSRetryAttempts* parameter is used with *PSRetryWait* (see below) to control retries of transmissions to Exchange. For example, the default settings:

```
PSRetryAttempts=3
PSRetryWait=15
```

... tell the program to retry a transmission to a maximum of three attempts (in addition to the initial attempt), at 15-second intervals.

If the error persists through all retry attempts, the program will note the error in the log, skip the current message property or element, and move on to process the next item. The retry attempts may appear in the program logs with no other documented error or warning.

Note that these *PSRetry* parameters can also be set in *adobjmerge.ini*.

## gwmigapp.ini [ActiveDirectory] PSRetryWait=

*integer; default = 15 or 40 (see below); new in ver 4.3*

Sets the interval (in seconds) between retry attempts when the program will retry data transmissions to the Exchange environment due to errors in prior attempts. This *PSRetryWait* parameter is used in conjunction with *PSRetryAttempts* to control retries of transmissions to Exchange, as described above for the *PSRetryAttempts* parameter. The default is 15 seconds for migration to a proprietary Exchange, or 40 seconds for migration to Office 365.

## gwmigapp.ini [ActiveDirectory] RemoveSrcAddresses=<#>

*boolean; default = 0*

Tells the program whether to remove all GroupWise source addresses (*gwise:proxy-Addresses*) so that mailbox-enabled accounts in Active Directory can be synchronized back to GroupWise as external users. If *RemoveSrcAddresses=0* (default), only the *gwise:uid-* address is removed. An object's *gwise:uid-* address must be removed once the object has been mailbox-enabled. Other GroupWise addresses are used to route mail via CMG's Mail Connector or Microsoft's Connector, if either is used, but are harmless to leave in place if neither connector is used.

## gwmigapp.ini [ActiveDirectory] SetCmgExcludeAttr=<#>

*boolean; default = 1*

Determines whether the Batch Migrator program will populate the extension attribute designated by the *CmgExcludeFromDirSync* parameter (also in this [Active Directory] section), which in turn specifies the AD attribute that will determine when merged and mailbox-enabled objects should synchronize back into GroupWise. This *SetCmgExcludeAttr* parameter is set to 1 (on) by default, but you can set *SetCmgExcludeAttr=0* to tell the program to not set (to ignore) the *CmgExcludeFromDirSync* parameter.

## gwmigapp.ini [ActiveDirectory] UseMapiRecipientFlag=<#>

*boolean; default = 1*

Determines whether the Admin-Driven Batch Migrator, when mailbox-enabling, will set the AD *mAPIRecipient* property according to the *MapiRecipientFlag* parameter value (also in this [ActiveDirectory] section). If *UseMapiRecipientFlag=0*, the program will leave the AD property at its *<not set>* default. But *UseMapiRecipientFlag=1* (the default) tells the program to set the AD property according to the *MapiRecipientFlag* parameter value. See the *MapiRecipientFlag* parameter above for more information about this feature.

**i** | **IMPORTANT:** The same two parameters must also be set to the same values in the [ActiveDirectory] section of *adobjmerge.ini*, so the Batch Migrator can correctly set the *mAPIRecipient* property to match the value set by the AD Object Merge Tool.

## gwmigapp.ini [ArchiveData] section

### gwmigapp.ini [ArchiveData]

Auto-set parameters — do not edit manually:

MigrateCalData=                      MigrateMailData=                      MigrateTrashFolder=

### gwmigapp.ini [ArchiveData] MigrateJunkFolder=<#>

*boolean; default = 1*

Determines whether the program will migrate the contents of the archived junk mail folder. *MigrateJunkFolder=0* tells the program to not migrate the junk folder. By default (*MigrateJunkFolder=1*), the junk folder will be migrated.

### gwmigapp.ini [ArchiveData] UsePseudoRoot=<#>

*Type: boolean; default: = 1; new in ver 4.3*

Determines whether the program will migrate archives to the primary Outlook mailbox (the main root) rather than to subfolders under a pseudo root. For example, *UsePseudoRoot=0* tells the program to migrate archives to the top-level mailbox (without a pseudo-root folder). By default (1), the program migrates archives to subfolders under a pseudo root.

## gwmigapp.ini [ErrorsToIgnore] section

### gwmigapp.ini [ErrorsToIgnore] Error[#]=<[#####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should ignore when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below).
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToIgnore]
Error0=[1465-32-4-00000000]
Error1=[7442-1-155-80072035]
Error2=[1400-31-108-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate the parameters from one another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## gwmigapp.ini [ErrorsToWarn] section

### gwmigapp.ini [ErrorsToWarn] Error[#]=<[####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should change to a warning when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToWarn]
Error1=[1465-32-4-00000000]
Error2=[7442-1-155-80072035]
Error3=[113-1-97-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0–99 inclusive.

## gwmigapp.ini [Exchange] section

### gwmigapp.ini [Exchange]

Auto-set parameters — do not edit manually:

AdminProfile=	NoServer=	ServerName=
CreateMailbox=	RmvForwarding=	SetForwarding=
ForwardingDomain=		

### gwmigapp.ini [Exchange] AddressListName=<NameOfList>

*string; default = All Users*

Specifies the name of the *identity* parameter for the *Update-AddressList* PowerShell cmdlet, which the batch migrator will run after mailbox-enabling a user-list *csv* file if *UpdateAddressList=1* (also in this [Exchange] section). See the *UpdateAddressList* parameter description for more information.

### gwmigapp.ini [Exchange] AddressPolicyName=<NameOfPolicy>

*string; default = Default Policy*

Specifies the name of the *identity* parameter for the *Update-EmailAddressPolicy* PowerShell cmdlet, which the Batch Migrator runs when mailbox-enabling a user-list *csv* file if *UpdateEmailAddressPolicy=1* (also in this [Exchange] section). See the *UpdateEmailAddressPolicy* parameter description for more information.

### gwmigapp.ini [Exchange] AdminAcctPoolFirstAccountSeq=<####>

*integer; default = 1001*

Specifies the account number you want to assign to the first admin in an Admin Account Pool group. This parameter combined with the *AdminAcctPoolRoleGroupName* parameter (see separate listing below) lets you configure account numbers for MFG's Admin Account Pool feature. (Admin Account Pooling is described in chapter 6 of the *MFG Administration Guide*.) For example:

```
[Exchange]
AdminAcctPoolRoleGroupName=QuestMFGAdminGroup
AdminAcctPoolFirstAccountSeq=2001
```

*RoleGroupName* is the name of the group whose first number you want to assign, while *FirstAccountSeq* is the number you want to assign to the first admin in the pool. Defaults are:

```
[Exchange]
AdminAcctPoolRoleGroupName=GW Migrator Admins Role Group
AdminAcctPoolFirstAccountSeq=1001
```

### **gwmigapp.ini [Exchange] AdminAcctPoolRoleGroupName=<GroupName>**

*string; default = GW Migrator Admins Role Group*

Specifies the name of an Admin Account Pool group for which you want to assign account numbers. This parameter combined with the *AdminAcctPoolFirstAccountSeq* parameter (see separate listing above) lets you configure account numbers for MFG's Admin Account Pool feature. (Admin Account Pooling is described in chapter 6 of the *MFG Administration Guide*.) For example:

```
[Exchange]
AdminAcctPoolRoleGroupName=QuestMFGAdminGroup
AdminAcctPoolFirstAccountSeq=2001
```

*RoleGroupName* is the name of the group whose first number you want to assign, while *FirstAccountSeq* is the number you want to assign to the first admin in the pool. Defaults are:

```
[Exchange]
AdminAcctPoolRoleGroupName=GW Migrator Admins Role Group
AdminAcctPoolFirstAccountSeq=1001
```

### **gwmigapp.ini [Exchange] AdminDomain=<WinDomain>**

*string; default = [none]*

The Windows domain required for the administrator's MAPI profile. Required for silent-mode operation. For example: *AdminDomain=sumptin.com*

### **gwmigapp.ini [Exchange] AdminPassword=<WinPassword>**

*string; default = [none]*

The Windows password required for the administrator's MAPI profile. Required for silent-mode operation. For example: *AdminPassword=Malcolm1863*

### **gwmigapp.ini [Exchange] AdminUsername=<WinUsername>**

*string; default = [none]*

The Windows username required for the administrator's MAPI profile. Required for silent-mode operation. For example: *AdminUsername=WilburBurns*

### **gwmigapp.ini [Exchange] ArchiveResolveAttendees=<#>**

*boolean; default = 1*

Tells the program whether to look up SMTP addresses in user archives in the Exchange GAL, to get Exchange addresses — e.g., so free-busy lookups will succeed. This feature is enabled by default, but can be disabled by setting *ArchiveResolveAttendees=0*. For example, to migrate data to PST files without connecting to the Exchange server, set:

```
ArchiveResolveAttendees=0
ServerMailResolveAttendees=0
```

... and also, in the [General] section:



```
[General]
PABDest=PST
ServerMailDest=PST
ArchiveDest=PST
```

## gwmigapp.ini [Exchange] AuthMechanismType=<keyword>

*keyword; default = kerberos*

Determines an authentication mechanism for remote PowerShell. The parameter value is a keyword that specifies the mechanism:

```
[Exchange]
AuthMechanismType=<keyword>
```

The default is *kerberos*, but you may specify another method for the keyword parameter value (case-insensitive). Valid values are:

basic	digest	negotiateWithImplicitCredential
credssp	kerberos	
default	negotiate	

These options are defined at [this Microsoft link](#).

## gwmigapp.ini [Exchange] DirSyncEnableRemoteMailbox=<#>

*boolean; default = 0*

This parameter and another (*LocalExchangeFederationServer*, also in this [Exchange] section) together support hybrid Office 365 migration (O365 provisioned by a local proprietary AD) with a local Exchange server and remote mailboxes (on O365) for some or most users. *DirSyncEnableRemoteMailbox* is a boolean parameter that, when enabled (1), tells the Data Migration Wizard to read the *LocalExchangeFederationServer* value to remote-mailbox-enable each principal provisioned during AD prep on the local Exchange. This feature therefore requires a valid FQDN value for *LocalExchangeFederationServer*, as described in that parameter entry below.

Note that this remote-mailbox-creation feature in preparing AD for AD sync does not support any settings associated with regular mailbox creation. For example, [Exchange] *EmailAddressPolicyEnabled* is not supported.

## gwmigapp.ini [Exchange] DeleteAutocompleteMessages=<#>

*boolean; default = 0*

Determines whether the Admin-Driven Batch Migrator will delete the *IPM.Configuration.Autocompletion* message from the Inbox at the end of a data migration run. For example:

```
[Exchange]
DeleteAutocompleteMessages=1
```

The AutoCompletion configuration message holds the cache of recently mailed-to users, but will cause errors in Outlook if the auto-completion lookup tables are corrupted—in which case Outlook will mistakenly tell users that valid addresses are invalid. The feature is off (0) by default.

## gwmigapp.ini [Exchange] DoGALLookupRetry=<#>

*boolean; default = 1*

Determines whether MFG's Admin-Driven Batch Migrator will skip MAPI retries for GAL lookups. Ordinarily in a GAL lookup the migration app will use the Exchange entry if it is found. If the lookup returns *no\_such\_user* the app will simply add it to the migrated item as a one-off SMTP address rather than a link to the target account. However, if the initial call returns any error, the app will retry (repeat the call). In some environments, the retry process can be time-consuming. In these instances, *DoGALLookupRetry=0* can be used to skip the retry. For example, *DoGALLookupRetry=0* tells the program to not retry a MAPI call



when searching the Exchange GAL if the initial call returns an error. By default (1) the program will retry such MAPI calls.

### gwmigapp.ini [Exchange] EmailAddressPolicyEnabled=<#>

*boolean; default = 1*

Determines how the program will set the Exchange option to *Automatically update e-mail addresses* based on e-mail address policy. (The *Automatically update...* feature is an Exchange feature, offered as a check box in the Exchange Management Console, on the *E-Mail Addresses* tab of the *Data Properties* for each mailbox.) This MFG *EmailAddressPolicyEnables* parameter specifies whether MFG should automatically enable or disable that Exchange feature for the AD objects in a designated user group. Example:

```
[Exchange]
EmailAddressPolicyEnabled=1
```

... tells MFG to enable the auto-update feature (mark the Exchange check box) for AD objects after processing. The alternative (0) disables the Exchange auto-updating feature (unmarks the Exchange check box). The default is 1 for any target type, including Office 365. **Note:** When mailbox-enabling, make sure that this *EmailAddressPolicyEnabled=<#>* parameter setting matches the setting for *adobjmerge.ini* [ActiveDirectory] *SetRecipPolicy=<#>*.

### gwmigapp.ini [Exchange] EwsRetryCount=<#>

### gwmigapp.ini [Exchange] EwsRetryWait=<##>

*integers (both); defaults = EwsRetryCount=3, EwsRetryWait=30*

Determines the maximum number of times that MFG's Admin-Driven Batch Migrator will retry a connection to Office 365 EWS autodiscover, in case of errors in the initial attempt, and sets the interval (in seconds) between retries. Some environments experience intermittent connection failures to O365, in which case MFG skips messages and users until the connection finally succeeds. This retry feature can make MFG more patient and "forgiving" of such connection failures with O365 EWS autodiscover. For example, the default settings:

```
[Exchange]
EwsRetryCount=3
EwsRetryWait=30
```

... tell the program to retry a connection to a maximum of three attempts (in addition to the initial attempt), at 30-second intervals. If the errors persist through all retry attempts, the program will note the failure in the log, skip the current message property or element, and move on to process the next item. Depending on the *LogLevel* setting, the retry attempts may appear in the program logs with no other documented error or warning.

### gwmigapp.ini [Exchange] IncludeGwiseProxyAddrs=<#>

*boolean; default = 0*

If BPOS-D is selected as the migration target, this *IncludeGwiseProxyAddrs* boolean parameter determines whether the Admin-Driven Batch Migrator will push GroupWise addresses into AD proxy addresses. By default (0) the Batch Migrator will not push GroupWise addresses into AD proxy addresses. But you can set *IncludeGwiseProxyAddrs=1* to let the Batch Migrator push GroupWise addresses into AD proxy addresses. Note that this parameter is ignored if the migration target is not BPOS-D.

### gwmigapp.ini [Exchange]

### LocalExchangeFederationServer=<LocalExchangeFQDN>

Type:	Default:	Occurs in:	New in:
string	[null]	Data Migr	4.7

*string; default = [null]*

This parameter and another (*DirSyncEnableRemoteMailbox*, also in this [Exchange] section) together support hybrid Office 365 migration (O365 provisioned by a local proprietary AD) with a local Exchange server and remote mailboxes (on O365) for some or most users. This *LocalExchangeFederationServer* value is a string for the FQDN of your local Exchange, which the program will need if *DirSyncEnableRemoteMailbox=1* to remote-mailbox-enable each principal provisioned during AD prep on the local Exchange.

AD credentials are used for this feature because MFG does not ask for separate credentials for the local Exchange. The AD admin must therefore be a member of Organization Management to run *EnableRemoteMailbox* on the local Exchange.

## gwmigapp.ini [Exchange] MAPIErrorsToRetry=<#####>

*string; default = 80040115; new in ver 3.6.2*

One of three parameters that together let you configure the Admin-Driven Batch Migrator to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MAPIErrorsToRetry* is used in conjunction with the *MessageRetryCount* and *MAPIRetryWait* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first, original attempt.)

For more information and an example of how this parameter works with the other two, see the *MessageRetryCount* parameter notes below.

## gwmigapp.ini [Exchange] MAPIRetryWait=<##>

*integer; default = 30; new in ver 3.6.2*

One of three parameters that together let you configure the Admin-Driven Batch Migrator to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MAPIRetryWait* is used in conjunction with the *MessageRetryCount* and *MAPIErrorsToRetry* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first, original attempt.)

For more information and an example of how this parameter works with the other two, see the *MessageRetryCount* parameter notes below.

## gwmigapp.ini [Exchange] MessageRetryCount=<#>

*integer; default = 3; new in ver 3.6.2*

One of three parameters that together let you configure the Admin-Driven Batch Migrator to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MessageRetryCount* is used in conjunction with the *MessageRetryWait* and *MAPIErrorsToRetry* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first, original attempt.)

If the call is still unsuccessful after the designated number of attempts, the program notes the error in the log, skips the current item, and moves on to process the next item. For example:

```
[Exchange]
MessageRetryCount=5
MAPIRetryWait=60
MAPIErrorsToRetry=8004010F,80040115
```

This example (retry 5 times, at 60-second intervals, for MAPI errors 8004010F and 80040115) would make the program more persistent and patient, and for an additional error, compared to the feature defaults (3 retries, every 30 seconds, for error 80040115 only). Depending on your setting for **Log level** (on the

Specify *Run Information* screen), the program's retry attempts may appear in the program logs with no other documented error or warning.

Prior releases did not handle these errors because they did not occur in traditional migrations to local Exchange servers, but they do occur now in migrations to remote "hosted Exchange" servers, such as Microsoft's BPOS. Since a series of retries is likely to "outlast" whatever condition caused the error, the MAPI call is much more likely to eventually succeed.

## **gwmigapp.ini [Exchange]**

### **O365AccountSkuld=<YOUR\_TENANT\_NAME:SKU\_PART\_NUMBER>**

*string; default = [none]*

Supports Office 365 licensing when organizations leverage multiple O365 account SKUs. If a *Multiple AccountSkus* error occurs, then the Batch Migrator has detected multiple accounts and requires that you specify one to use for this feature. The *Get-MSOLAccountSKU* cmdlet can retrieve the *AccountSkulds* for all valid licenses. Review the list and determine the appropriate SKU for the current licensing operation. Then specify the SKU you want to use in this new parameter in the [Exchange] section of *gwmigapp.ini*:

```
[Exchange]
O365AccountSkuld=<YOUR_TENANT_NAME:SKU_PART_NUMBER>
```

### **gwmigapp.ini [Exchange] O365UsageLocation=<xx>**

*keyword; default = [none]*

The mandatory two-letter *Usage Location* code that Microsoft requires for its Office 365 licenses (per [this Microsoft article](#)). The two-letter keyword must conform to the standardized values for [ISO 3166-1-alpha-2](#).

### **gwmigapp.ini [Exchange] PstFileName=<filename>**

*string; default = [none]*

Specifies a common output PST filename for all migrating users, when migrating data to PSTs. This value is used in conjunction with the *PSTDir* column in the user list *csv* file, which specifies a unique PST directory for each user. The combination of a common *PstFileName* value with unique *PSTDir* values assures that the newly created PST files will be identifiable by user. Example:

```
PstFileName=migr-archive.pst
```

... combined with *PSTDir* column values set to `\\server\home\user1`, `user2`, `user3`, etc., would produce these new PST files in these locations:

```
\\server\home\user1\migr-archive.pst
\\server\home\user2\migr-archive.pst
\\server\home\user3\migr-archive.pst
... [etc.]
```

Otherwise, if the *PSTDir* column is unspecified and the *PstFileName* parameter sets the output filename to a common value, the multiple files saved to the same directory would be differentiated only by appended digits: *migr-archive 1.pst*, *migr-archive 2.pst*, *migr-archive 3.pst*, and so forth. Each user in a user-list *csv* file should therefore have a unique directory specified in the *PSTDir* column, such as the user's home directory on a server.

If *PSTDir* columns are specified but the *PstFileName* setting is left unspecified, the new PST filenames are derived from users' display names.

### **gwmigapp.ini [Exchange] PurgePABS=<#>**

*boolean; default = 0*

*PurgePABS=1* tells the program to clear the contents of Personal Address Books in Exchange before migrating corresponding PABs from GroupWise—useful if, for example, you had already migrated PABs, but then wanted to migrate them again. The default *PurgePABS=0* disables the feature, so no PABs are

cleared. This feature clears only Exchange PABs that correspond to GroupWise PABs about to be migrated, and will not clear any other existing Exchange PABs.

### **gwmigapp.ini [Exchange] RemigrateMode=<#>**

*keyword; default = 1*

Tells the program what to do when an item that has already been migrated to Exchange is queued to migrate again. Valid parameter values are 0, 1 and 2, corresponding to these program settings:

0 = **always replace**: Deletes the Exchange copy of the item and then remigrate it from GroupWise.

1 = **never replace**: The source item in GroupWise will not be migrated, leaving the previously migrated copy unupdated in Exchange.

2 = **replace on change**: Replaces a previously migrated message only if the message on the source has been changed since the time of migration.

### **gwmigapp.ini [Exchange] ServerMailResolveAttendees=<#>**

*boolean; default = 1*

Tells the program whether to look up SMTP addresses in server mail in the Exchange GAL, to get Exchange addresses — e.g., so free-busy lookups will succeed. This feature is enabled by default, but can be disabled by setting the parameter to 0. For example, to migrate data to PST files without connecting to the Exchange server, set parameters as shown in the example above for the *ArchiveResolveAttendees* parameter.

### **gwmigapp.ini [Exchange] SetResourceToAutoAccept=<#>**

*boolean; default = 1*

Enables or disables the Exchange autoaccept feature for resources. (Autoaccept is an Exchange feature, available only when migrating to Exchange 2013 or 2010, or Office 365.) For example, *SetResourceToAutoAccept=1* (the default) tells the Batch Migrator to enable Exchange's autoaccept feature when creating resource mailboxes. To disable the feature, set *SetResourceToAutoAccept=0*.

### **gwmigapp.ini [Exchange] ShowResolveAttendeesOpt=<#>**

*boolean; default = 0*

Determines whether the two **Resolve attendees...** check box fields will appear in the *Office 365 Mail Settings* screen of the Admin-Driven Batch Migrator program. The check boxes are hidden (*ShowResolveAttendeesOpt=0*) by default, in which case the features will be enabled (as if both check boxes were marked).

### **gwmigapp.ini [Exchange] UpdateAddressList=<#>**

*boolean; default = 0*

*UpdateAddressList=1* tells the Batch Migrator to run the *Update-AddressList* PowerShell cmdlet after mailbox-enabling a collection, to complete the AD provisioning of recipients who are only partially provisioned. *Update-AddressList* will run with its identity parameter set to "All Users" by default, or to some other identity value if set by the *AddressListName* parameter (also in this [Exchange] section). This can be useful when objects have been provisioned by tools such as Microsoft Identity Integration Server (MIIS) or custom provisioning tools that do not fully provision recipient objects. By default (0), the cmdlet will not run.

### **gwmigapp.ini [Exchange] UpdateEmailAddressPolicy=<#>**

*boolean; default = 0*

*UpdateEmailAddressPolicy=1* tells the Batch Migrator to run the *Update-EmailAddressPolicy* PowerShell cmdlet after mailbox-enabling a collection, to complete the AD provisioning of recipients who are only partially provisioned. The *Update-EmailAddressPolicy* cmdlet will run with its identity parameter set to "Default Policy" by default, or to some other identity value if set by the *AddressPolicyName* parameter (also

in this [Exchange] section). This can be useful when objects have been provisioned by tools such as Microsoft Identity Integration Server (MIIS) or custom provisioning tools that do not fully provision recipient objects. By default (0), the cmdlet will not run.

### gwmigapp.ini [Exchange] ValidateForwardingAddresses=<#>

*boolean; default = 0; new in ver 4.5.1*

Determines whether the program, when migrating to Office 365, will (1) or will not (0) validate forwarding addresses when forwarding mail from GroupWise to Office 365. The option is relevant only if G-to-E mail forwarding is enabled for O365, and only if a *csv* field other than *targetAddress* has been selected (in the *Specify Source-to-Target Mail Forwarding Domain* screen) for forwarding.

This parameter corresponds to the **Validate mail-forwarding addresses** check box on the *Office 365 Mail Settings* screen. MFG writes this value to the INI file from the GUI entry; Quest recommends you not manually enter or change this value except at the direction of Quest Support.

### gwmigapp.ini [Exchange] VerifyO365AdminRights=<#>

*boolean; default = 0*

Determines whether the program, when migrating to Office 365, will bypass the *Get-MailBoxPermission* and *Add-MailBoxPermission* checks, and assume that it already has the necessary rights to proceed with the migration. This can be useful if, due to security standards and/or regulations, you prefer to not give the MFG migration account *Organization Management* rights to your entire O365 environment. By default (*VerifyO365AdminRights=1*) the program will perform the *Get-MailBoxPermission* and *Add-MailBoxPermission* checks.

### gwmigapp.ini [Exchange] WriteMessageRetryWait=<##>

*integer; default = 30; new in ver 3.6.2*

One of two parameters that let you configure the Admin-Driven Batch Migrator to retry the migration of an entire message when an error causes the overall message to fail. This *WriteMessageRetryWait* parameter tells the program how long to wait (in seconds) between successive attempts to migrate a message. See the *WriteMessageTryCount* parameter notes (below) for more information and an example.

### gwmigapp.ini [Exchange] WriteMessageTryCount=<#>

*integer; default = 1; new in ver 3.6.2*

One of two parameters that let you configure the Admin-Driven Batch Migrator to retry the migration of an entire message when some error causes the overall message to fail. For example:

```
[Exchange]
WriteMessageTryCount=3
WriteMessageRetryWait=60
```

This *WriteMessageTryCount* parameter tells the program how many times to try to migrate a message, at intervals specified by the *WriteMessageRetryWait* value (number of seconds), before giving up the attempts. If a message fails again on the last attempt, the program notes the failure in its log and resumes processing with the next message. By default (*WriteMessageTryCount=1*) the program will try only once (and not retry at all).

This feature can complement the MAPI-error retry feature, which permits retries at the MAPI-error level (see the *MessageRetryCount* parameter above, also in this [Exchange] section). At the MAPI-error level, if a MAPI error remains unresolved after the specified number of retry attempts, the program notes the error in its log and resumes processing with the next element or property of the message. But if an error causes an entire message to fail, these message-retry parameters let you tell the program to retry processing the entire message, from the beginning, at specified intervals, through a specified number of iterations.

## gwmigapp.ini [Filter] section

### gwmigapp.ini [Filter]

Auto-set parameters — do not edit manually:

AttachSize=	FirstCalDate=	FirstMailDate=
	LastCalDate=	LastMailDate=

## gwmigapp.ini [Filters] section

### gwmigapp.ini [Filters] Filter[#]=<AlphanumericString>

*string; default = [as noted below]*

Each line specifies a folder, by name, to be excluded from the migration. Upon installation, *gwmigapp.ini* contains these *Filter[#]* parameters by default:

filter0=(\$Alarms)	filter12=(\$MAPIIPM Subtree)
filter1=(Group Calendars)	filter13=(\$MAPIOutbox)
filter2=(Rules)	filter14=(\$MAPISent)
filter3=(\$Design)	filter15=(\$MAPITrash)
filter4=Alarms	filter16=(Discussion Threads)
filter5=(CalSummary)	filter17=(\$ToDo)
filter6=(~MAPI\$P(Internal))	filter18=(\$FolderInfo)
filter7=(IPMCOMMONVIEWS)	filter19=(\$POP3)
filter8=(IPMVIEWS)	filter20=(To do's\By Category)
filter9=(Search Root)	filter21=(To do's\By Status)
filter10=(\$MAPIInbox)	filter22=(Mail Threads)*
filter11=(\$MAPIInfo)	filter23=(\$FolderAllInfo)

You can add to this list to define more folders to be excluded. Note that the digits appended to the parameter key names need not appear in numerical order, but must in aggregate form an uninterrupted numerical sequence and must include *Filter0*.

## gwmigapp.ini [FolderNames] section

### gwmigapp.ini [FolderNames] ~JUNKMAIL=JUNK

*keyword; default = [none]*

The *JUNK* parameter value is a keyword that tells the program to create a folder named "Junk E-mail," which prevents Outlook 2003 from automatically creating its own *Junk E-mail* folder upon start-up. For example: *~JUNKMAIL=JUNK*

Some admins prefer not letting Outlook create the folder, since the auto-generated folder would inherit the ACL from the root, which can exceed the privileges that they want to allow for this folder. This is only an issue with Outlook 2003.

# gwmigapp.ini [General] section

## gwmigapp.ini [General]

Auto-set parameters — do not edit manually:

ArchiveDest=	Migrate=	RunName=
DoGroupWiseAdmin=	MigrateArchives=	ServerMailDest=
DoTargetAdmin=	MigrateFreqContacts=	SkippedMsgCnt=
EndTime=	MigratePAB=	SMTPTranslationTable=
ErrorCnt=	MigrateServerMail=	StartTime=
FileAsFormat=	MigrationID=	Status=
FilteredApptCnt=	MovedApptCnt=	ThreadCnt=
FilteredAttachCnt=	MovedApptSize=	TotalApptCnt=
FilteredAttachSize=	MovedContactCnt=	TotalContactCnt=
FilteredMsgCnt=	MovedMsgCnt=	TotalMsgCnt=
ForceTargetAdmin=	MovedMsgSize=	UseLargePsts=
GroupsToProvisionPath=	PABDest=	UserDoneCnt=
KeyError=	PdlsDoneCnt=	UserErrList=
LicenseKey=	PdlsFailCnt=	UserFailCnt=
LogDir=	Provider=	UserList=
LogLevel=	ProvisionGroups=	WarningCnt=

## gwmigapp.ini [General] ACLRootVisibleOnly=<#>

*boolean; default = 1; new in ver 3.5*

Tells the program whether to migrate the ACLs of shared-access folders when *MigrateEmptyFolders=0* (in the [GroupWise] section, telling the program to not migrate empty folders). If *MigrateEmptyFolders=1*, the ACLs are migrated and any *ACLRootVisibleOnly* parameter is ignored. The program will migrate shared-folder ACLs by default (*ACLRootVisibleOnly=1*), but you can disable the feature by *ACLRootVisibleOnly=0*.

## gwmigapp.ini [General] ACLs=<#>

*boolean; default = 1*

*ACLs=0* tells the program to not copy ACL information to Exchange during the migration. By default (1) the program *will* copy ACL information and will set folder permissions in Exchange.

## gwmigapp.ini [General] ArchiveFolderName=<alphanum>

*string; default = GroupWise ArchiveNew in ver 4.1*

Specifies the name of the root folder in Outlook that MFG will create to receive migrated archives. This folder is named GroupWise Archive by default (if this parameter is unspecified).

## gwmigapp.ini [General] AttachmentRetryCount=<##>

*integer; default = 0*

Specifies the number of additional times the program should attempt to migrate an attachment if the first attempt fails. Valid values are integers in the range 0–99 inclusive. For example, *AttachmentRetryCount=3* tells the program that, if an attachment fails to migrate, the program should retry the migration three more times before giving up.

## gwmigapp.ini [General] CommandLine=<alphanum>

*string; default = [none]*

Forces the GroupWise client/API to access and migrate remote mailboxes instead of accounts on the server, by these parameters:



For GroupWise 5.5.x: `CommandLine=/ps-C:\Novell\GroupWise\remote`

For GroupWise 6.x: `CommandLine=/pr-C:\Novell\GroupWise\remote`

By default (if this parameter is omitted), the GroupWise client/API will access and migrate data from accounts on the server.

### **gwmigapp.ini [General] DoInactivityTimer=<#>**

*boolean; default = 1*

Determines whether the Batch Migrator program's inactivity timer is enabled or disabled. For example, `DoInactivityTimer=1` tells the program to abort a process if it receives no response from the destination server for longer than a specified number of minutes following a data transmission. The timeout limit is 40 minutes by default, or you can specify some other value by the `InactivityTimeoutMinutes` parameter (also in this [General] section). To disable this feature, set `DoInactivityTimer=0`.

### **gwmigapp.ini [General] DoMigrateNotesOfContacts=<#>**

*boolean; default = 0; new in ver 4.5.1*

Tells the Admin-Driven Batch Migrator to migrate (1) or not migrate (0) contacts' notes when migrating PABs. The option is off (0) by default. This feature is also available for the SSDM, where it is enabled/disabled (independently) by the same parameter in `gwdtapp.ini`.

### **gwmigapp.ini [General] EnableBPOS=<#>**

*boolean; default = 0*

`EnableBPOS=1` enables the BPOS features in the Batch Migrator, and adds `BPOS-D` to the drop-down list of eligible migration targets. This feature is off (0) by default.

### **gwmigapp.ini [General] ForwardingMethod=<#>**

*keyword; default = [varies]*

Determines which method the program will use for Active Directory mail forwarding. The keyword value is an integer that must be one of these three valid values:

0: Forward by creating a hidden contact and attaching it as the alternate recipient of the mailbox.

1: Forward by setting the `targetAddress` attribute (default when migrating to Hosted Exchange).

2: Forward by SMTP address via PowerShell (default and mandatory when migrating to Office 365, also available for Exchange 2010).

### **gwmigapp.ini [General] InactivityTimeoutMinutes=<##>**

*integer; default = 40*

Sets the inactivity timeout limit, by number of minutes, if the timeout feature is enabled by `DoInactivityTimer=1` (see separate parameter above in this same [General] section). The inactivity timeout feature tells the program to abort a process if it receives no response from the destination server for longer than a specified number of minutes following a data transmission. For example, `InactivityTimeoutMinutes=60` tells the program to set the timeout threshold at 60 minutes (1 hour).

### **gwmigapp.ini [General]MaxPstSize=<#####>**

*integer; default = 1500000*

The maximum size (in kilobytes) of PSTs, beyond which the program will split the file into two or more files. Valid values are integers  $0 \leq x \leq 2097152$ . For example: `MaxPstSize=625000`



## **gwmigapp.ini [General] MergeAddressBookFolder=<alphanum>**

*string; default = Contacts*

The name of the single folder to which the program will merge multiple address books, if you want to migrate them to a folder other than the default Contacts folder. For example, *MergeAddressBookFolder=AllContacts*

This parameter is ignored if *MergeAddressBooks=0* and no address-book value is specified for the *MergeAddressBookList* parameter (see below).

## **gwmigapp.ini [General] MergeAddressBookList=<Name1; Name2; ...>**

*string; default = [none]*

Tells the program to merge the contents of the named address book(s) into the single Contacts folder, or into a single folder as named by the *MergeAddressBookFolder* parameter (see above). Address-book name values are both case- sensitive and space-sensitive, and you may specify two or more by using semicolons to separate the folder names in the list. This parameter is ignored if *MergeAddressBooks=1* (see below), which tells the program to migrate all address books into a single folder. By default the program migrates different address books to different folders, one folder per address book.

## **gwmigapp.ini [General] MergeAddressBooks=<#>**

*boolean; default = 0*

*MergeAddressBooks=1* tells the program to merge the contents of all migrated address books into the single Contacts folder, or into a single folder as named by the *MergeAddressBookFolder* parameter (see above). By default the program migrates different address books to different folders, one folder per address book.

## **gwmigapp.ini [General] MaxDiscrepancyThreshold=<####>**

*integer; default = 5000*

At the end of the migration for a user, MFG compares the number of the estimated items that to be migrated with those that GroupWise has provided in the progress. If the difference reaches this threshold, MFG notes an error in the log file.

## **gwmigapp.ini [General] RetryMax=<#>**

*integer; default = 10*

When the migration program exits due to an unexpected error or exception, *RetryMax* controls the maximum retry count. *RetryMax=0* tells the program to disable retries.

## **gwmigapp.ini [General] SkipIncomingSharedFolder=<#>**

*boolean; default = 0*

*SkipIncomingSharedFolder=1* tells the program to skip the incoming shared folder during the migration.

## **gwmigapp.ini [General] Silent=<#>**

*boolean; default = 0*

*Silent=1* tells the program to run in Silent mode — to use the configuration options provided during a previous session and run without user input.

## **gwmigapp.ini [General] UseSeparatePSTs=<#>**

*boolean; default = 1*

*UseSeparatePSTs=0* tells the program to migrate all data into a single *pst* file per user. The default is 1, so data is migrated into multiple *pst* files.

## gwmigapp.ini [General] DoCategories=<#>

*boolean; default = 1; new in ver 3.6.1*

Determines whether the Admin-Driven Batch Migrator will migrate message *Categories*—color differentiation of messages within a folder, at the user's designation. The feature is enabled (*DoCategories=1*) by default, but can be disabled by *DoCategories=0*. If the GroupWise source *Category* color does not exactly match one of Outlook's 25 standard colors, MFG will assign it to the closest matching color.

**Note:** This feature requires GroupWise 6.5.7 or later on the admin's migration workstation.

## gwmigapp.ini [GroupWise] section

### gwmigapp.ini [GroupWise]

Auto-set parameters — do not edit manually:

ArchiveDir=	ForwardingGwDomain=	ProxyMode=
AutoReplySubject=	ForwardingGwPo=	ProxyPassword=
AutoReplyText=	ForwardingSMTPDomain=	ProxyUserid=
DomainPath=	FwdCsvField=	SetAutoReply=
DoPassword=	GWIAPrefix=	SetDelItemsRule=
DoRmvForwarding=	InternetAddressingEnabled=	SetPurgeRule=
DoSetForwarding=	NewPassword=	UseApiGateway=
DoVisibility=	PasswordMode=	VisibilityMode=
ForwardingDomain=	PostOffice=	

### gwmigapp.ini [GroupWise] AdminApiLockWaitSecs=<###>

*integer; default = 300; new in ver 3.1.0.19*

Number of seconds the program will wait before attempting (again) to set a user password, if the first or previous attempt was unsuccessful. For example, *AdminApiLockWaitSecs=300* (the default) tells the program to wait five minutes (300 seconds) before each retry to set a user password.

### gwmigapp.ini [GroupWise] AdminApiRetries=<###>

*integer; default = 0; new in ver 3.1.0.19*

Number of times the program will attempt to set a user password before giving up (if all attempts are unsuccessful). For example, *AdminApiRetries=6* tells the program to retry setting a user password up to six times before giving up. The default setting of 0 tells the program to never retry; if the first attempt is unsuccessful, the program will simply log the error and move on to the next user.

### gwmigapp.ini [GroupWise] AlwaysExpandRecips=<#>

*boolean; default = 0; new in ver 3.6*

Determines whether the program, when migrating a message sent to a distribution list, will migrate only the name of the list as the recipient, or will preserve the original GroupWise recipients by expanding the list prior to migration—to record all list members as of the *Send* date. For example, *AlwaysExpandRecips=1* tells the program to expand distribution lists prior to migration. If this feature is off (0), the program migrates only the name of a list, and a later expansion of the list name (in Outlook) will yield an inaccurate list of recipients if list membership had changed after the message was originally sent.

The size of an expanded recipient list may be limited by the *MaxExpandedRecips* parameter (see separate listing below), also in this [GroupWise] section.

## **gwmigapp.ini [GroupWise] DebugFolder=<FolderName>**

*string; default = [none]*

The name of a single source folder containing all the data to be migrated in this program run. All other source folders are filtered for non-migration (useful for troubleshooting). For example:

*DebugFolder=Cabinet*

## **gwmigapp.ini [GroupWise] ExportHTML=<#>**

*boolean; default = 1*

*ExportHTML=0* tells the program to not migrate HTML message bodies. By default (*ExportHTML=1*) the program will migrate HTML message bodies.

## **gwmigapp.ini [GroupWise] ExportRTF=<#>**

*boolean; default = 0*

*ExportRTF=1* tells the program to migrate RTF message bodies. By default (*ExportRTF=0*) the program will not migrate RTF message bodies.

## **gwmigapp.ini [GroupWise] ForwardLockWaitSecs=<###>**

*integer; default = 300; new in ver 3.1.0.19*

Number of seconds the program will wait before attempting (again) to set mail-forwarding for a particular user, if the first or previous attempt was unsuccessful. For example, *ForwardLockWaitSecs=300* (the default) tells the program to wait five minutes (300 seconds) before each retry to set a user's mail-forwarding.

## **gwmigapp.ini [GroupWise] ForwardRetries=<##>**

*integer; default = 0; new in ver 3.1.0.19*

Number of times the program will attempt to set a user's mail-forwarding before giving up (if all attempts are unsuccessful). For example, *ForwardRetries=6* tells the program to retry setting a user's mail-forwarding up to six times. The default (0) tells the program to never retry; if the first attempt is unsuccessful, the program simply logs the error and moves on to the next user.

## **gwmigapp.ini [GroupWise] GWSoapUrl=<AlphanumericString>**

*string; default = [none]*

This parameter corresponds to a GUI element in the Directory Exporter: the **GroupWise Soap Server URL** field in the *GroupWise System Information* screen. The value specifies the location of the GroupWise SOAP web service. This parameter is optional, but will help resolve a duplicate-meetings issue that can occur in some scenarios in some environments where Quest CMG coexistence product is also deployed. When a GroupWise user creates a meeting in GroupWise, and then changes it (or changes one instance of a recurring series) after he/she is migrated, the invitees' calendars may show duplicate instances unless the SOAP service is enabled and its URL is specified. This solution works only for GroupWise 8 and later (not for GroupWise 7).

MFG writes this value to the INI files from an admin's GUI entry; Quest recommends you not manually enter or change this value except at the direction of Quest Support.

## gwmigapp.ini [GroupWise] LoginXxxxx=

*integer; default = varies*

These four *Login* parameters let you delay the program's login to GroupWise:

- **LoginAttempts=<####>**

*integer; default = 1*

In the event of a login failure, number of times ( $0 \leq x \leq 9999$ ) to retry. For example: *LoginAttempts=3*

- **LoginDelay=<####>**

*integer; default = 0*

Number of seconds ( $0 \leq x \leq 9999$ ) to delay before logging in. For example: *LoginDelay=10*

- **LoginRetryWait=<####>**

*integer; default = 0 (no wait)*

Number of seconds ( $0 \leq x \leq 9999$ ) to wait between login attempts, if *LoginAttempts* > 1. For example: *LoginRetryWait=10*

- **LoginWait=<####>**

*integer; default = 300 (5 minutes)*

Number of seconds ( $0 \leq x \leq 9999$ ) to wait for the login mutex. For example: *LoginWait=600*

## gwmigapp.ini [GroupWise] MaxExpandedRecips=<####>

*integer; default = 500*

Specifies the maximum number of recipients that MFG will expand from a recipient list, when *AlwaysExpandRecips=1* (see "[gwmigapp.ini \[GroupWise\] AlwaysExpandRecips=<#>](#)" on page 42.). That is, if the full list is longer than the specified max, MFG will truncate the expanded list at this specified limit. The value may be set to any positive 32-bit integer, and defaults to 500, and should always match the maximum recipient list size in Exchange (which is 500 by default).

## gwmigapp.ini [GroupWise] MigrateDocRefs=<#>

*boolean; default = 0*

*MigrateDocRefs=1* tells the program to migrate documents and document references as attachments. By default, documents and document references do not migrate. Note that if any of these documents are GroupWise hidden messages, you must also set [GroupWise] *MigrateHiddenMsg=1* (see below) if you want to migrate the hidden messages.

## gwmigapp.ini [GroupWise] MigrateEmptyFolders=<#>

*boolean; default = 1*

*MigrateEmptyFolders=0* tells the program to not migrate empty folders. By default, the program will migrate all folders regardless of whether they contain any items. **Note:** If *MigrateEmptyFolders=0*, the program migrates ACLs by default unless *AcIRootVisibleOnly=0* (in the [General] section).

## gwmigapp.ini [GroupWise] MigrateEmptySharedFolders=<#>

*boolean; default = 0; new in version 3.5*

*MigrateEmptySharedFolders=1* tells the program to migrate empty shared folders, if *MigrateEmptyFolders=0* (which tells the program to *not* migrate other empty folders). For example:

```
MigrateEmptyFolders=0
MigrateEmptySharedFolders=1
```

... tells the program to migrate empty shared folders, but to *not* migrate other (non-shared) empty folders. By default (*MigrateEmptySharedFolders=0*), empty shared folders are not migrated. If *MigrateEmptyFolders=1* (the default), the program ignores this *MigrateEmptySharedFolders* parameter and migrates all empty folders.

### **gwmigapp.ini [GroupWise] MigrateHiddenMsg=<#>**

*boolean; default = 0*

*MigrateHiddenMsg=1* tells the program to migrate GroupWise hidden messages that occur as Document References in mail folders. This feature is associated with a check box (**Migrate Document References in mail folders by migrating a copy of the document**) on the *Specify Data for Migration* screen of the program. If that check box is marked, and if the documents to be migrated are GroupWise hidden messages, the hidden messages will be migrated only if *MigrateHiddenMsg=1*.

### **gwmigapp.ini [GroupWise] MigrateMySubjectAppointments=<#>**

*boolean; default = 0*

*MigrateMySubjectAppointments=1* tells the program to migrate the My Subject field of an appointment from GroupWise, and use it as the appointment subject in the target.

### **gwmigapp.ini [GroupWise] MigrateMySubjectDocs=<#>**

*boolean; default = 0*

*MigrateMySubjectDocs=1* tells the program to migrate the My Subject field of a document from GroupWise, and use it as the document subject in the target.

### **gwmigapp.ini [GroupWise] MigrateMySubjectEmails=<#>**

*boolean; default = 0*

*MigrateMySubjectEmails=1* tells the program to migrate the My Subject field of an email from GroupWise, and use it as the email subject in the target.

### **gwmigapp.ini [GroupWise] MigrateMySubjectNotes=<#>**

*boolean; default = 0*

*MigrateMySubjectPhoneMsgs=1* tells the program to migrate the My Subject field of a note from GroupWise, and use it as the note subject in the target.

### **gwmigapp.ini [GroupWise] MigrateMySubjectPhoneMsgs=<#>**

*boolean; default = 0*

*MigrateMySubjectPhoneMsgs=1* tells the program to migrate the My Subject field of a phone message from GroupWise, and use it as the message subject in the target.

### **gwmigapp.ini [GroupWise] MigrateMySubjectTasks=<#>**

*boolean; default = 0*

*MigrateMySubjectTasks=1* tells the program to migrate the My Subject field of a task from GroupWise, and use it as the task subject in the target.

### **gwmigapp.ini [GroupWise] MigrateNotesAs=<#>**

*keyword; default = 3*

Determines whether the program will migrate GroupWise Reminder Notes as tasks or all-day events. By default (3), the program migrates Reminder Notes as all-day events. Valid keyword values are single digits 1, 2 or 3 only:

- 1: migrate as a task
- 2: migrate as either all-day event or task (see below)
- 3: migrate as an all-day event

If *MigrateNotesAs=2*, the program looks at the due-date property to determine whether it is an all-day event (due-date property is set to 0), or a task (due-date property is set to anything *other than* 0).

### **gwmigapp.ini [GroupWise] MigrateSharedFolders=<#>**

*boolean; default = 0*

*MigrateSharedFolders=1* tells the program to migrate an independent copy of each shared folder to each user to whom the source folder was shared. By default the program migrates only a single copy of each shared folder to the user who owned it in GroupWise, and sets ACL permissions on that folder for other users to whom the GroupWise folder was shared.

### **gwmigapp.ini [GroupWise] MigrateSharedPABs=<#>**

*boolean; default = 0*

*MigrateSharedPABs=1* tells the program to migrate an independent copy of each shared PAB to each user to whom the source PAB was shared. By default the program migrates only a single copy of each shared PAB to the user who owned it in GroupWise, and sets ACL permissions on that PAB for other users to whom the GroupWise PAB was shared.

### **gwmigapp.ini [GroupWise] MsgIdxEnd=<###>**

*integer; default = [no limit]*

The highest message number you want to migrate (if you want to migrate only messages up to and including a particular message number. For example, *MsgIdxEnd=6108* tells the program to migrate messages up to and including message index #6108. To migrate all messages up to the latest message number, simply omit this item from the INI file. Valid values are 0 up to the number of messages in the message store.

### **gwmigapp.ini [GroupWise] MsgIdxStart=<###>**

*integer; default = [no limit]*

The lowest message number you want to migrate (if you want to migrate only messages after a particular message number). Valid values are 0 up to the number of messages in the message store. This can be helpful if the program crashes unexpectedly and you want to resume the migration after the message that caused the crash. For example *MsgIdxStart=1421* tells the program to migrate messages beginning with index #1421. To migrate all messages back to the earliest, omit this parameter from the INI file.

### **gwmigapp.ini [GroupWise] PabsToIgnore=<PABName>**

*string (length must not exceed 8191 chars); default = [none]*

Specifies one or more shared address books that the Batch Migrator will not be permitted to migrate. This parameter lets you disallow the migration of shared address books reserved for use by Quest CMG Directory Connector:

```
[Groupwise]
PabsToIgnore=<PabName>
```

(Many MFG sites also use Quest CMG for coexistence, and CMG's Directory Connector requires one or more GroupWise shared address books to serve as repositories for objects copied by the Directory Connector from Active Directory. The Directory Connector address book/s should not be migrated.)

To specify two or more PABs to exclude, separate their names with semicolons:

```
PabsToIgnore=Name1;Name2;Name3
```

## gwmigapp.ini [GroupWise] <PostOffice>.<Domain>=<<IP>:<Port>>

*string; default = [none]; new in ver 4.5.1*

Associates a particular GroupWise Post Office/domain with its corresponding *IPAddress:port*. These parameters support unambiguous GroupWise-to-Exchange mail forwarding for two users sharing the same name and Object ID, but residing in different Post Offices and different domains. You can configure as many of these parameters as necessary to associate a unique *IPAddress:port* with each GW Post Office/domain. For example, if:

- John Doe is in 192.168.0.1:1677, with an address of *jdoe@hokum.sitraka1.com*, and
- a different John Doe is in 192.168.0.2:1688 with an address of *jdoe@sitraka2.com*

... then you could define:

```
[GroupWise]
hokum_mail.hokum=192.168.0.1:1677
sitraka2_mail.sitraka2=192.168.0.2:1688
```

These settings would forward the mail for each John Doe (and any other user pairs sharing the same name and Object ID) to the appropriate target address in Exchange.

## gwmigapp.ini [GroupWise] ProxyAccountRights=<#>

*boolean; default = 0*

*ProxyAccountRights=1* tells the program to migrate users' proxy rights along with the users' other account information. Prior to version 3.1, the program migrated proxy rights by default. As of version 3.1, the default is to *not* migrate proxy rights (*ProxyAccountRights=0*).

## gwmigapp.ini [GroupWise] RetryCount=<#>

*integer; default = 3; new in ver 3.6.2*

One of two parameters that let you configure the Admin-Driven Batch Migrator to retry the migration of an entire message when an error occurs in the connection between MFG and the GroupWise server. If a Groupwise-MFG connection error causes an entire message to fail, these message-retry parameters let you tell the program to retry processing the entire message, from the beginning, at specified intervals, through a specified number of iterations. For example:

```
[GroupWise]
RetryCount=5
RetryDelay=60
```

... tells the program to retry 5 times (6 total tries), at intervals of 60 seconds, and then give up if still unsuccessful. The *RetryCount* tells the program how many times to retry to migrate a message (if not successful on the first attempt), at intervals specified by the *RetryDelay* parameter, before giving up the attempts. The defaults are *RetryCount=3* and *RetryDelay=30*. If a message fails again on the last attempt, the program notes the failure in its log and resumes processing with the next message.

(The *WriteMessageTryCount* parameter, in the [Exchange] section, does the same thing for errors that occur between MFG and Exchange.)

## gwmigapp.ini [GroupWise] RetryDelay=<##>

*integer; default = 30; new in ver 3.6.2*

One of two parameters that let you configure the Admin-Driven Batch Migrator to retry the migration of an entire message when an error occurs in the connection between MFG and the GroupWise server. If a Groupwise-MFG connection error causes an entire message to fail, these message-retry parameters let you tell the program to retry processing the entire message, from the beginning, at specified intervals, through a specified number of iterations. See the *RetryCount* parameter notes (above, also in this [GroupWise] section) for more information and an example.



## gwmigapp.ini [GroupWise] SetDelSentItemsRule=<#>

*boolean; default = 1*

Determines whether the Batch Migrator will purge the GroupWise *Sent Items* folder of forwarded items. For example, if the Batch Migrator is set to forward mail, then

```
SetDelSentItemsRule=1
```

tells the program to set an additional rule to delete forwarded items from the GroupWise *Sent Items* folder, and to automatically remove the rule when GroupWise forwarding is removed. The feature is enabled (*SetDelSentItemsRule=1*) by default.

## gwmigapp.ini [GroupWise] SkipCountByFolder=<#>

*boolean; default = 1*

*SkipCountByFolder=1* tells the program to not try to count messages by folder, but instead just count all messages. If *SkipCountByFolder=0* the program counts by folder and then by message to work around a bug in the GroupWise API that limits the folder count to 8191 messages. If a folder contains more than 8191 messages, then the program will automatically switch methods. If the program crashes when estimating the number of messages to be migrated, setting this parameter value to 1 will probably preempt the crash.

## gwmigapp.ini [GroupWise] UseTrustedAPI=<#>

*boolean; default = 1*

*UseTrustedAPI=0* tells the program to not use Novell's Trusted Application API feature to automatically register itself as a trusted application, so it can migrate GroupWise user data without user passwords or proxy authorization. By default, the program will use the Trusted Application API feature (*UseTrustedAPI=1*).

# gwmigapp.ini [Outlook] section

## gwmigapp.ini [Outlook]

Auto-set parameters — do not edit manually:

```
PstRootDir= PstDirMode=
```

# gwmigapp.ini [PowerShell] section

## gwmigapp.ini [PowerShell] IdleConnectionTimeoutSeconds=<##>

*integer; default = 30*

Specifies how long (in seconds) MFG will hold open an idle remote PowerShell connection before closing it. The default *IdleConnectionTimeoutSeconds=30* will be suitable for most environments. Note that this feature applies only to the duration of an idle state during a connection. Each command execution resets this timer to zero, so a series of commands with only short idle periods between commands could keep the connection open indefinitely. *IdleConnectionTimeoutSeconds=0* would tell MFG to not wait (wait 0 seconds) for a second command after a first, so every PowerShell connection would close immediately after only one command.

Remember that this parameter must be set on each migration machine, and should also be set in *adobjmerge.ini*.



## gwmigapp.ini [PowerShell] MaxPowerShellConnections=<#>

*integer; default = 0*

Specifies a migration-server-wide limit for the number of concurrent PowerShell connections MFG can open. This parameter is intended to eliminate the possibility of MFG exceeding the Microsoft-imposed limit of 9 simultaneous runspace connections (remote PowerShell). The recommended value is derived by:

$$(R - 1) / S$$

... where *R* is the number of simultaneous runspaces allowed by your tenant (typically 9), and *S* is the number of migration servers you will use. For example, if your limit is 9 runspaces and you intend to use one migration server, then *MaxPowerShellConnections=8 = (9-1)/1*. Or for a 9-runspace limit with 2 migration machines: *MaxPowerShellConnections=4 = (9-1)/2*. The default *MaxPowerShellConnections=0* is interpreted as “no limit”, effectively turning off this limiting feature.

Remember that this parameter must be set on each migration machine, and should also be set in *adobjmerge.ini*.

## gwmigapp.ini [PowerShell] PropagationDelay=<##>

*integer; default = 40*

Determines how long (in seconds) the Admin-Driven Batch Migrator will wait after setting admin permissions on an Office 365 mailbox before proceeding with the migration. For example, the default *PropagationDelay=40* tells the program to wait 40 seconds before proceeding. This can be helpful to resolve permissions issues resulting from an O365 Powershell-Exchange propagation delay.

# gwmigapp.ini [ServerData] section

## gwmigapp.ini [ServerData]

Auto-set parameters — do not edit manually:

MigrateCalData=                      MigrateMailData=                      MigrateTrashFolder=

## gwmigapp.ini [ServerData] MigrateJunkFolder=<#>

*boolean; default = 1*

*MigrateJunkFolder=1* (the default) tells the program to migrate the contents of the server junk mail folder. *MigrateJunkFolder=0* tells the program to not migrate the junk folder.

# gwmigapp.ini [Status] section

## gwmigapp.ini [Status]

Auto-set parameters — do not edit manually:

Code=                                      Text=                                      Info=

# gwmigapp.ini [X400Domains] section

## gwmigapp.ini [X400Domains] Domain[#]=<DomainName>

*string; default = [none]*

The list of X400 gateway domains in use for GroupWise, for example:

```
Domain0=x400DomainA  
Domain1=x400DomainB  
Domain2=x400DomainC  
Domain3=x400DomainD
```

These parameters let the program convert GroupWise X400 addresses into Exchange X400 addresses, since GroupWise and Exchange use different formats for X400 addressing.

# Parameters for the SSDM

The gwdtapp.ini (SSDM) has the following sections:

- gwdtapp.ini [ArchiveData] section
- gwdtapp.ini [ErrorsToIgnore] section
- gwdtapp.ini [Exchange] section
- gwdtapp.ini [Filter] section
- gwdtapp.ini [Filters] section
- gwdtapp.ini [FolderNames] section
- gwdtapp.ini [General] section
- gwdtapp.ini [GroupWise] section
- gwdtapp.ini [PAB] section
- gwdtapp.ini [Screens] section
- gwdtapp.ini [ServerData] section

## gwdtapp.ini [ArchiveData] section

### gwdtapp.ini [ArchiveData] MigrateCalData=<#>

*boolean; default = 1*

Determines whether the program will migrate archived calendar data. *MigrateCalData=0* tells the program to *not* migrate archived calendar data. By default (*MigrateCalData=1*), the program *will* migrate archived calendar data.

### gwdtapp.ini [ArchiveData] MigrateJunkFolder=<#>

*boolean; default = 1*

Determines whether the program will migrate the contents of the archived junk mail folder. *MigrateJunkFolder=0* tells the program to not migrate the junk folder. By default (*MigrateJunkFolder=1*), the junk folder will be migrated.

### gwdtapp.ini [ArchiveData] MigrateTrashFolder=<#>

*boolean; default = 0*

*MigrateTrashFolder=1* tells the program to migrate the trash folder if the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default. A value of 0 tells the program to *not* migrate the trash folder, or sets the associated checkbox to be unmarked by default. The most common use of this parameter is to provide the necessary entry value when the program is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to other parameters used to customize the SSDM.

# gwdtapp.ini [ErrorsToIgnore] section

## gwdtapp.ini [ErrorsToIgnore] Error[#]=<[#####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should ignore when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToIgnore]
Error0=[1465-32-4-00000000]
Error1=[7442-1-155-80072035]
Error2=[1400-31-108-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0-99 inclusive.

# gwdtapp.ini [ErrorsToWarn] section

## gwdtapp.ini [ErrorsToWarn] Error[#]=<[#####-###-###-#####]>

*string; default = [none]*

Each *Error[#]* parameter specifies a single particular runtime error that the program should change to a warning when documenting its activities and outcomes in its log file. Each parameter value:

- Must conform to the form shown here, consisting entirely of digits and hyphens enclosed within square brackets (see example below); **and**
- Must be the full, exact string of a single defined error code (no wildcards or substrings).

For example:

```
[ErrorsToWarn]
Error1=[1465-32-4-00000000]
Error2=[7442-1-155-80072035]
Error3=[113-1-97-00000000]
```

The digit(s) appended to an *Error[#]* key name arbitrarily differentiate one parameter from another. Multiple *Error[#]* parameters need not appear in numerical order, need not form an uninterrupted numerical sequence, and need not begin at 0. But the program will read only *Error[#]* parameters whose differentiating digits are within the range 0-99 inclusive.

# gwdtapp.ini [Exchange] section

## gwdtapp.ini [Exchange] ArchiveResolveAttendees=<#>

*boolean; default = 1*

Tells the program whether to look up SMTP addresses in user archives in the Exchange GAL, to get Exchange addresses—e.g., so free-busy lookups will succeed. This feature is enabled by default, but can be disabled by setting the parameter to 0. For example, to migrate data to PST files without connecting to the Exchange server, set:

```
ArchiveResolveAttendees=0
ServerMailResolveAttendees=0
```

... and also, in the [General] section:

```
[General]
PABDest=PST
ServerMailDest=PST
ArchiveDest=PST
```

## gwdtapp.ini [Exchange] AuthMechanismType=<keyword>

*keyword; default = kerberos*

Determines an authentication mechanism for remote PowerShell. The parameter value is a keyword that specifies the mechanism:

```
[Exchange]
AuthMechanismType=<keyword>
```

The default is kerberos, but you may specify another method by entering the name as the parameter value (case-insensitive). Valid keyword values are:

basic	digest	negotiateWithImplicitCredential
credssp	kerberos	
default	negotiate	

These options are defined at [this Microsoft link](#).

## gwdtapp.ini [Exchange] DeleteAutocompleteMessages=<#>

*boolean; default = 0*

Determines whether the SSDM will delete the *IPM.Configuration.Autocompletion* message from the Inbox at the end of a data migration run. For example:

```
[Exchange]
DeleteAutocompleteMessages=<#>
```

The AutoCompletion configuration message holds the cache of recently mailed-to users, but will cause errors in Outlook if the auto-completion lookup tables are corrupted—in which case Outlook will mistakenly tell users that valid addresses are invalid. The feature is off (0) by default.

## gwdtapp.ini [Exchange] DoGALLookupRetry=<#>

*boolean; default = 1*

Determines whether the SSDM will skip MAPI retries for GAL lookups. Ordinarily in a GAL lookup the migration app will use the Exchange entry if it is found. If the lookup returns *no\_such\_user* the app will simply add it to the migrated item as a "one-off" SMTP address rather than a link to the target account. However, if the initial call returns any error, the app will retry (repeat the call). In some environments, the retry process can be time-consuming. In these instances, *DoGALLookupRetry=0* can be used to skip the retry. For example:

```
DoGALLookupRetry=0
```

... tells SSDM to not retry a MAPI call when searching the Exchange GAL if the initial call returns an error. By default (1) the program will retry such MAPI calls.

## gwdtapp.ini [Exchange] MAPIErrorsToRetry=<#####>

*string; default = 80040115*

One of three parameters that together let you configure the SSDM to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MAPIErrorsToRetry* is used in conjunction with the *MessageRetryCount* and *MAPIRetryWait* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first, original attempt.)

For more information and an example of how this parameter works with the other two, see the *MessageRetryCount* parameter notes below.

## gwdtapp.ini [Exchange] MAPIRetryWait=<##>

*integer; default = 30*

One of three parameters that together let you configure the SSDM to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MAPIRetryWait* is used with the *MessageRetryCount* and *MAPIErrorsToRetry* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first attempt.)

For more information and an example of how this parameter works with the other two, see the *MessageRetryCount* parameter notes below.

## gwdtapp.ini [Exchange] MessageRetryCount=<#>

*integer; default = 3**New in ver 3.6.2*

One of three parameters that together let you configure the SSDM to retry MAPI calls that return MAPI errors typically caused by network latency issues and/or poor bandwidth. *MessageRetryCount* is used in conjunction with the *MessageRetryWait* and *MAPIErrorsToRetry* parameters (also in this [Exchange] section): The program will retry the call if its preceding attempt returns any of the error codes specified by the *MAPIErrorsToRetry* parameter, at intervals specified (in seconds) by *MAPIRetryWait*, as long as the number of prior retries does not exceed the *MessageRetryCount* value. ("Retries" do not include the first, original attempt.)

If the call is still unsuccessful after the designated number of attempts, the program notes the error in the log, skips the current item, and moves on to process the next item. For example:

```
[Exchange]
MessageRetryCount=5
MAPIRetryWait=60
MAPIErrorsToRetry=8004010F,80040115
```

This example (retry 5 times, at 60-second intervals, for MAPI errors 8004010F and 80040115) would make the program more persistent and patient, and for an additional error, compared to the feature defaults (3 retries, every 30 seconds, for error 80040115 only). Depending on your setting for **Log level** (on the *Specify Run Information* screen), the program's retry attempts may appear in the program logs with no other documented error or warning.

Prior releases did not handle these errors because they did not occur in traditional migrations to local Exchange servers, but they do occur now in migrations to remote "hosted Exchange" servers, such as Microsoft's BPOS. Since a series of retries is likely to "outlast" whatever condition caused the error, the MAPI call is much more likely to eventually succeed.

## gwdtapp.ini [Exchange] RemigrateMode=<#>

*keyword; default = 1*

Tells the program what to do when an item that has already been migrated to Exchange is queued to migrate again. The parameter value is a single digit corresponding to these program settings:

- **0 = always replace:** deletes the Exchange copy of the item and then remigrate it from GroupWise.
- **1 = never replace:** the source item in GroupWise will not be migrated, leaving the previously migrated copy unupdated in Exchange.
- **2 = replace on change:** replaces a previously migrated message only if the message on the source has been changed since the time of migration.

## gwdtapp.ini [Exchange] ServerMailResolveAttendees=<#>

*boolean; default = 1*

Tells the program whether to look up SMTP addresses in server mail in the Exchange GAL, to get Exchange addresses—e.g., so free/busy lookups will succeed. This feature is enabled by default, but can be disabled by setting the parameter to 0. For example, to migrate data to PST files without connecting to the Exchange server, set parameters as shown in the example above (for the *ArchiveResolveAttendees* parameter).

# gwdtapp.ini [Filter] section

## gwdtapp.ini [Filter]

Auto-set parameters — do not edit manually:

AttachSize=

FirstCalDate=

FirstMailDate=

LastCalDate=

LastMailDate=

# gwdtapp.ini [Filters] section

## gwdtapp.ini [Filters] Filter[#]=<AlphanumericString>

*string; default = [as noted below]*

Each line specifies a folder, by name, to be excluded from the migration. Upon installation, *gwdtapp.ini* contains these *Filter[#]* parameters by default:

filter0=(\$Alarms)	filter12=(\$MAPIIPM Subtree)
filter1=(Group Calendars)	filter13=(\$MAPIOutbox)
filter2=(Rules)	filter14=(\$MAPISent)
filter3=(\$Design)	filter15=(\$MAPITrash)
filter4=Alarms	filter16=(Discussion Threads)
filter5=(CalSummary)	filter17=(\$ToDo)
filter6=(~MAPI\$P(Internal))	filter18=(\$FolderInfo)
filter7=(IPMCOMMONVIEWS)	filter19=(\$POP3)
filter8=(IPMVIEWS)	filter20=(To do's\By Category)
filter9=(Search Root)	filter21=(To do's\By Status)
filter10=(\$MAPIInbox)	filter22=(Mail Threads)*
filter11=(\$MAPIInfo)	filter23=(\$FolderAllInfo)

You can add to this list to define more folders to be excluded. The digits appended to parameter key names need not appear in numerical order, but must in aggregate form an uninterrupted numerical sequence and must include *Filter0*.

# gwdtapp.ini [FolderNames] section

## gwdtapp.ini [FolderNames] ~JUNKMAIL=JUNK

*keyword; default = [none]*

The *JUNK* parameter value is a keyword that tells the program to create a folder named *Junk E-mail*, which will prevent Outlook 2003 from automatically creating its own *Junk E-mail* folder upon start-up. For example:

```
~JUNKMAIL=JUNK
```

Some admins prefer not letting Outlook create the folder, since the auto-generated folder would inherit the ACL from the root, which can exceed the privileges they want to allow for this folder. This is only an issue with Outlook 2003.

# gwdtapp.ini [General] section

## gwdtapp.ini [General]

Auto-set parameters — do not edit manually:

AskDestClientType=	LicenseKey=	MigrationID=
DestClientType=	Provider=	SMTPTranslationTable=

## gwdtapp.ini [General] ACLs=<#>

*boolean; default = 1*

Determines whether the program will copy ACL information to Exchange during the migration. By default (*ACLs=1*) the SSDM will copy ACL information and will set folder permissions in Exchange.

## gwdtapp.ini [General] AnyPst=<value>

This is a temporary "scratch" value entered and used by the program. Do not manually edit this value except at the direction of Quest support staff.

## gwdtapp.ini [General] AppDoesArch=<#>

## gwdtapp.ini [General] AppDoesMail=<#>

## gwdtapp.ini [General] AppDoesPabs=<#>

*boolean; default = 1 (for each of the three)*

Determines whether the corresponding migration option will be visible to the user on the *Specify Data for Migration* screen. The default *AppDoesXxxx=1* makes the option visible. *AppDoesXxxx=0* makes the option invisible (unavailable to the user) on the screen, and also:

- masks the portions of the *Welcome* and *Migration Report* screens that pertain to the hidden option; and
- disallows the migration of that data type regardless of other parameter settings elsewhere in the INI file (that is, this parameter would override *MigrateArchives=1*).

Note that these parameters do not set the marked-vs-unmarked status of the corresponding checkboxes, although checkbox status can be controlled by the *MigrateXxxxx* parameters. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to others used to customize the SSDM.



## gwdtapp.ini [General] ArchiveDest=<destination>

*keyword* (valid values are *server* or *pst*); default = *PST*

See [General] *ArchiveDestServerArchive* below.

## gwdtapp.ini [General] ArchiveDestServerArchive=<#>

*boolean*; default = 0

The *ArchiveDest* parameter specifies whether Archives should be migrated to the target server (*ArchiveDest=server*), or to a PST file (*ArchiveDest=pst*). The only valid values are *pst* (the default) or *server*.

If *ArchiveDest=server*, the SSDM then reads another boolean parameter (*ArchiveDestServerArchive=<#>*) to determine whether Archives should be sent to the user's server-mail mailbox (0) or to a personal-archive mailbox (1):

```
[General]
ArchiveDest=Server
ArchiveDestServerArchive=1
```

The default is *ArchiveDestServerArchive=0*, so that Archives will be migrated to the user's server-mail mailbox.

The Exchange personal-archives feature, available only in Exchange 2010 and later (including Office 365), lets a user keep archived items in a separate mailbox where they are more secure and accessible than in PST files. The *ArchiveDestServerArchive* parameter is ignored when migrating to an Exchange target where the personal archives feature is not available.

This table shows how the parameters should be set to achieve each of the three possible destinations:

<u>To server-mail mailbox:</u>	<u>To personal-archive mailbox:</u>	<u>To pst file:</u>
[General] ArchiveDest=Server ArchiveDestServerArchive=0	[General] ArchiveDest=Server ArchiveDestServerArchive=1	[General] ArchiveDest=pst

The boolean *ArchiveDestServerArchive* parameter is relevant only if *ArchiveDest=server*. If *ArchiveDest=pst* then the *ArchiveDestServerArchive* parameter is ignored.

## gwdtapp.ini [General] AskFilter=<#>

*boolean*; default = 0

Enables or disables date filtering, which lets you limit the items migrated by date: no earlier than, no later than, or both. The feature is disabled (0) by default—i.e., the date-filtering options are not presented to the user. **Note:** To use date filters with the SSDM, you may need to register the *mscal.ocx* file.

## gwdtapp.ini [General] AttachmentRetryCount=<##>

*integer*; default = 0

Specifies the number of additional times (valid range is 0–99 inclusive) to attempt to migrate an attachment if the first attempt fails. For example, *AttachmentRetryCount=3* tells the SSDM that, if an attachment fails to migrate, the program should retry the migration three more times before giving up.

## gwdtapp.ini [General] DoCategories=<#>

*boolean*; default = 1 *New in ver 3.6.1*

Determines whether the SSDM will migrate message *Categories* (color differentiation of messages within a folder, at the user's designation). The feature is enabled by default, but can be disabled by *DoCategories=0*. If the GroupWise source Category color does not exactly match one of Outlook's 25 standard colors, MFG will assign it to the closest matching color. **Note:** This feature requires GroupWise 6.5.7 or later on the admin's migration workstation.

## **gwdtapp.ini [General] DoMigrateNotesOfContacts=<#>**

*boolean; default = 0; new in ver 4.5.1*

Tells the SSDM to migrate (1) or not migrate (0) contacts' notes when migrating PABs. The option is off (0) by default. This feature is also available for the Admin-Driven Batch Migrator, where it is enabled/disabled (independently) by the same parameter in *gwmigapp.ini*.

## **gwdtapp.ini [General] LogLevel=<#>**

*keyword; default = 1*

Specifies how much detail the program will include in its run logs. Valid values are single-digit integers in the range 0–3 inclusive. The default *LogLevel=1* is suitable for most routine production runs. If you are having trouble and want to diagnose a problem, set the value to 3 (verbose): *LogLevel=3*

## **gwdtapp.ini [General] MaxPstSize=<#####>**

*integer; default = 1500000*

The maximum size (in kilobytes) of PSTs, beyond which the program will split the file into two or more files. Valid values are integers in the range 0–20971520 inclusive. For example: *MaxPstSize=625000*

## **gwdtapp.ini [General] MaxDiscrepancyThreshold=<####>**

*integer; default = 5000*

At the end of the migration for a user, MFG compares the number of the estimated items that to be migrated with those that GroupWise has provided in the progress. If the difference reaches this threshold, MFG notes an error in the log file.

## **gwdtapp.ini [General] MergeAddressBookFolder=<alphanum>**

*string; default = Contacts*

The name of the single folder to which the program will merge multiple address books, if you want to migrate them to a folder other than the default *Contacts* folder. For example:

```
MergeAddressBookFolder=AllContacts
```

This parameter is ignored if *MergeAddressBooks=0* and no address-book value is specified for the *MergeAddressBookList* parameter.

## **gwdtapp.ini [General] MergeAddressBookList=<Name1; Name2; ...>**

*string; default = [none]*

Tells the program to merge the contents of the named address book(s) into the single *Contacts* folder, or into a single folder as named by the *MergeAddressBookFolder* parameter (see above). Address-book name values are both case- sensitive and space-sensitive. You may specify two or more by using semicolons to separate the folder names in the list.

This parameter will be ignored if *MergeAddressBooks=1*, which tells the program to migrate all address books into a single folder (as described below). By default the program will migrate different address books to different folders, one folder per address book.

## **gwdtapp.ini [General] MergeAddressBooks=<#>**

*boolean; default = 0*

```
MergeAddressBooks=1
```

... tells the program to merge the contents of all migrated address books into the single *Contacts* folder, or into a single folder as named by the *MergeAddressBookFolder* parameter. By default the program will migrate different address books to different folders, one folder per address book.

## gwdtapp.ini [General] MigrateArchives=<#>

*boolean; default = 0*

MigrateArchives=1

... tells the program to migrate archives if the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default. A value of 0 tells the program to not migrate archives, or sets the associated checkbox to be unmarked by default. This parameter is ignored if *AppDoesArch=0*. The most common use of this parameter is to provide the necessary entry value when the program is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to others used to customize the SSDM.

## gwdtapp.ini [General] MigrateFreqContacts=<#>

*boolean; default = 0*

*MigrateFreqContacts=1* tells the program to migrate Frequent Contacts if the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default. A value of 0 tells the program to not migrate Frequent Contacts, or sets the associated checkbox to be unmarked by default. The most common use of this parameter is to provide the necessary entry value when the program is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to other parameters used to customize the SSDM.

## gwdtapp.ini [General] MigratePAB=<#>

*boolean; default = 0*

*MigratePAB=1* tells the program to migrate PABs if the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default. If *MigratePAB=0* (default) the program will not migrate PABs, or will set the associated checkbox to be unmarked by default. This parameter is ignored if *AppDoesPabs=0*. This parameter is most commonly used to provide the necessary entry value when SSDM is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to others used to customize the SSDM.

## gwdtapp.ini [General] MigrateServerMail=<#>

*boolean; default = 0*

*MigrateServerMail=1* tells the program to migrate server mail if the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default. A value of 0 tells the program to not migrate server mail, or sets the associated checkbox to be unmarked by default. This parameter is ignored if *AppDoesMail=0*. The most common use of this parameter is to provide the necessary entry value when the program is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to other parameters used to customize the SSDM.

## gwdtapp.ini [General] PABDest=<destination>

*keyword (server or pst); default = PST*

See [General] *PABDestServerArchive* below.

## gwdtapp.ini [General] PABDestServerArchive=<#>

*boolean; default = 0*

The *PABDest* parameter specifies whether PABs should be migrated to the target server (*PABDest=server*), or to a PST file (*PABDest=pst*). The only valid values are *pst* (the default) or *server*.

If *PABDest=server*, the SSDM then reads a separate boolean parameter (*PABDestServerArchive=<#>*) to determine whether PABs should be sent to the user's server-mail mailbox (0) or to a personal-archive mailbox (1):

```
[General]
PABDest=Server
PABDestServerArchive=1
```

The default is *PABDestServerArchive=0*, so that PABs will be migrated to the user's server-mail mailbox.

Personal archives is a feature only in Exchange 2010 and later, including Office 365, that lets a user keep archived items in a separate mailbox—where they are more secure and accessible than in PST files. The *PABDestServerArchive* parameter is ignored when migrating to a non-Office 365 hosted-Exchange environment, where personal archives are not available.

This table shows how the parameters should be set to achieve each of the three possible destinations:

<u>To server-mail mailbox:</u>	<u>To personal-archive mailbox:</u>	<u>To pst file:</u>
[General] PABDest=Server PABDestServerArchive=0	[General] PABDest=Server PABDestServerArchive=1	[General] PABDest=pst

The boolean *PABDestServerArchive* parameter is relevant only if *PABDest=server*. If *PABDest=pst* then the *PABDestServerArchive* parameter is ignored.

### gwdtapp.ini [General] SelectedProfile=<ProfileName>

*string; default = [none]*

Specifies the Outlook user profile to which the program will migrate data. For example:  
*SelectedProfile=Joel Brennen*

The program needs this *SelectedProfile* parameter only if more than one eligible profile will be found on any user's workstation. The SSDM chapter of the MFG *Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to other parameters used to customize the SSDM.

### gwdtapp.ini [General] SelectedPstDir=<path>

*string; default = [Outlook default directory]*

Specifies the destination folder (path) for the user's migrated archive. The *<path>* is specified as a full UNC or mapped-drive path (without filename) relative to the user's computer. For example:

```
SelectedPstDir=c:\Program Files\Outlook\Archive
```

The default is the Outlook default directory, which the program can determine from its operating environment. This parameter is therefore necessary only if you want users' migrated archives to go somewhere other than the Outlook default directory. The SSDM chapter of the MFG *Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to others used to customize the SSDM.

### gwdtapp.ini [General] ServerMailDest=<destination>

*keyword; default = PST*

See *ServerMailDestServerArchive* below.

### gwdtapp.ini [General] ServerMailDestServerArchive=<destination>

*boolean; default = 0*

The *ServerMailDest* parameter specifies whether ServerMail should be migrated to the target server (*ServerMailDest=server*), or to a PST file (*ServerMailDest=pst*). The only valid values are *pst* (the default) or *server*.

If *ServerMailDest=server*, the SSDM then reads a separate boolean parameter (*ServerMailDestServerArchive=<#>*) to determine whether ServerMail should go to the user's server-mail mailbox (0) or to a personal-archive mailbox (1):

```
ServerMailDest=Server
ServerMailDestServerArchive=1
```

The default is *ServerMailDestServerArchive=0*, so that ServerMail will be migrated to the user's server-mail mailbox.

Personal archives is a feature in Exchange 2010 and Office 365 (only) that lets a user keep archived items in a separate mailbox—where they are more secure and accessible than PST files. The *ServerMailDestServerArchive* parameter is ignored when migrating to a non-Office 365 hosted- Exchange environment, where the personal archives feature is not available.

This table shows how the parameters should be set to achieve each of the three possible destinations:

<u>To server-mail mailbox:</u>	<u>To personal-archive mailbox:</u>	<u>To pst file:</u>
[General] ServerMailDest=Server ServerMailDestServerArchive=0	[General] ServerMailDest=Server ServerMailDestServerArchive=1	[General] ServerMailDest=pst

Note that the boolean *ServerMailDestServerArchive* parameter is relevant only if *ServerMailDest=server*. If *ServerMailDest=pst* then the *ServerMailDestServerArchive* parameter is ignored.

## gwdtapp.ini [General] UseLargePsts=<#>

*boolean; default = 0*

Determines whether the program will use large PSTs. By default (1) the program will use large PSTs if possible. Note: A setting of *UseLargePsts=1* will not force Unicode PSTs if Outlook 2003 is not installed. The program cannot create PSTs without the proper MAPI components (from Outlook 2003) in place.

# gwdtapp.ini [GroupWise] section

## gwdtapp.ini [GroupWise] CommandLine=<alphanum>

*string; default = [none]*

Forces the GroupWise client/API to access and migrate remote mailboxes instead of accounts on the server, by these parameters:

For GroupWise 5.5.x:

```
CommandLine=/ps-C:\Novell\GroupWise\remote
```

For GroupWise 6.x:

```
CommandLine=/pr-C:\Novell\GroupWise\remote
```

By default (if this parameter is omitted), the GroupWise client/API will access and migrate data from accounts on the server.

## gwdtapp.ini [GroupWise] DebugFolder=<FolderName>

*string; default = [none]*

The name of a single source folder containing all the data to be migrated in this program run. All other source folders are filtered for non-migration (useful for troubleshooting). For example:

```
DebugFolder=Cabinet
```

## gwdtapp.ini [GroupWise] GWSoapUrl=<AlphanumericString>

*string; default = [none]*

This parameter corresponds to a GUI element in the Directory Exporter: the **GroupWise Soap Server URL** field in the *GroupWise System Information* screen. The value specifies the location of the GroupWise SOAP web service. This parameter is optional, but will help resolve a duplicate-meetings issue that can occur in some scenarios in some environments where Quest CMG coexistence product is also deployed. When a GroupWise user creates a meeting in GroupWise, and then changes it (or changes one instance of a recurring series) after he/she is migrated, the invitees' calendars may show duplicate instances unless the SOAP service is enabled and its URL is specified. This solution works only for GroupWise 8 and later (not for GroupWise 7).

MFG writes this value to the INI files from an admin's GUI entry; Quest recommends you not manually enter or change this value except at the direction of Quest Support.

## gwdtapp.ini [GroupWise] MigrateDocRefs=<#>

*boolean; default = 0*

Tells the program whether to migrate documents and document references, as attachments. By default (*MigrateDocRefs=0*), documents and document references do *not* migrate.

## gwdtapp.ini [GroupWise] MigrateEmptyFolders=<#>

*boolean; default = 1*

Tells the program whether to migrate empty folders. By default (*MigrateEmptyFolders=1*), the program will migrate all folders regardless of whether they contain any items. **Note:** The migration program will not migrate ACLs if *MigrateEmptyFolders=0*.

## gwdtapp.ini [GroupWise] MigrateNotesAs=<#>

*keyword; default = 3*

Determines whether the program will migrate GroupWise Reminder Notes as tasks or all-day events. By default (3), the program migrates Reminder Notes as all-day events. Valid keyword values are single digits 1, 2 or 3 only:

- 1: migrate as a task
- 2: migrate as either all-day event or task (see below)
- 3: migrate as an all-day event

If *MigrateNotesAs=2*, the program looks at the due-date property to determine whether it is an all-day event (if the due-date property is set to 0), or a task (if the due-date property is set to anything *other than* 0).

## gwdtapp.ini [GroupWise] MigrateSharedFolders=<#>

*boolean; default = 0*

Tells the program to migrate an independent copy of each shared folder to each user to whom the source folder was shared (*MigrateSharedFolders=1*), or migrate only a single copy of each shared folder to the user who owned it in GroupWise, and set ACL permissions on that folder for other users to whom the GroupWise folder was shared (0).

## gwdtapp.ini [GroupWise] MigrateSharedPABs=<#>

*boolean; default = 0*

Tells the program to migrate an independent copy of each shared PAB to each user to whom the source PAB was shared (*MigrateSharedPABs=1*), or migrate only a single copy of each shared PAB to the user who owned it in GroupWise, and set ACL permissions on that PAB for other users to whom the GroupWise PAB was shared (0).

## gwdtapp.ini [GroupWise] MsgIdxEnd=<###>

*integer; default = no limit*

Specifies the highest message number you want to migrate (if you want to migrate only messages up to and including a particular message number). For example, *MsgIdxEnd=6108* tells the program to migrate messages up to and including index #6108. To migrate *all* messages up to the latest, simply omit this parameter from the INI file. Valid values are integers from 0 up to the number of messages in the message store.

## gwdtapp.ini [GroupWise] MsgIdxStart=<###>

*integer; default = no limit*

Specifies the lowest message number you want to migrate (if you want to migrate only messages with message numbers equal to or higher than a particular value). For example, *MsgIdxStart=1421* tells the program to migrate only messages with index #1421 or higher. To migrate *all* messages up to the latest, simply omit this parameter from the INI file. Valid values are integers from 0 up to the number of messages in the message store. This parameter can be helpful if the program crashes unexpectedly and you want to resume the migration after the message that caused the crash. To migrate all messages beginning with the earliest, omit this parameter from the INI file.

## gwdtapp.ini [GroupWise] PabsToIgnore=<PABName>

*string (length must not exceed 8191 chars); default = [none]*

Specifies one or more shared address books that the SSDM will not be permitted to migrate. This parameter lets you disallow the migration of shared address books reserved for use by Quest CMG Directory Connector:

```
[Groupwise]
PabsToIgnore=<PabName>
```

(Many MFG sites also use Quest CMG for coexistence, and CMG's Directory Connector requires one or more GroupWise shared address books to serve as repositories for objects copied by the Directory Connector from Active Directory. The Directory Connector address book/s should not be migrated.)

To specify two or more PABs to exclude, separate their names with semicolons:

```
PabsToIgnore=Name1;Name2;Name3
```

## gwdtapp.ini [GroupWise] ProxyAccountRights=<#>

*boolean; default = 0*

Tells the program whether to migrate users' proxy rights along with the users' other account information. For example, *ProxyAccountRights=1* tells the program to migrate proxy rights. Prior to version 3.1, the program migrated proxy rights by default. As of version 3.1, the default is to not migrate proxy rights (*ProxyAccountRights=0*).

## gwdtapp.ini [GroupWise] SkipCountByFolder=<#>

*boolean; default = 1*

Tells the program whether to try to count messages by folder (*SkipCountByFolder=0*), or instead just count all messages (1). If *SkipCountByFolder=0* the program counts by folder and then by message to work around a bug in the GroupWise API that limits the folder count to 8191 messages. If a folder contains more than 8191 messages, then the program will automatically switch methods. If the program crashes when estimating the number of messages to be migrated, setting *SkipCountByFolder=1* will probably preempt the crash.



# gwdtapp.ini [PAB] section

## gwdtapp.ini [PAB] MakeDefault=<#>

*boolean; default = 0*

Tells the program whether to make the migrated Contacts folder the default address book (*MakeDefault=1*). The default (0) leaves this option disabled.

## gwdtapp.ini [PAB] ShowFirst=<#>

*boolean; default = 0*

Tells the program whether to set the migrated Contacts folder to be the first address book shown in Outlook (*ShowFirst=1*). The default (0) leaves this option disabled.

# gwdtapp.ini [Screens] section

## For all parameters in the gwdtapp.ini [Screens] section ...

See the SSDM chapter of the MFG *Scenarios Guide* for an explanation of how these parameters relate to one another to customize the SSDM.

## gwdtapp.ini [Screens] Finished=<alphanumeric>

*keyword; default = show*

Determines whether, or under what circumstances, the *Migration Report* screen (the *Finished* screen) will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run). For example, *Finished=skip* tells the program to display the *Finished* screen only if the program encounters errors during the run, which would require a decision as to whether to view the *Error Log Report*.

## gwdtapp.ini [Screens] MigrateWhat=<alphanumeric>

*keyword; default = show*

Determines whether, or under what circumstances, the *Specify Data for Migration* screen will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run). For example, *MigrateWhat=skip* tells the program to display the screen only if the information it collects cannot be determined by other entries or the operating environment.

## gwdtapp.ini [Screens] Profile=<alphanumeric>

*keyword; default = skip*

Determines whether, or under what circumstances, the *Select Profile* screen will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run). For example, *Profile=skip* tells the program to display the *Select Profile* screen only if the information it collects cannot be determined by other entries or the operating environment.

## gwdtapp.ini [Screens] Progress=<alphanumeric>

*keyword; default = show*



Determines whether, or under what circumstances, the *Migrating Data* screen (the "Progress" screen) will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run) — but since this screen does not collect any information, the skip and silent options are equivalent. For example, *Progress=skip* tells the program to display the *Migrating Data* screen only if the information it collects cannot be determined by other entries or the operating environment — so this screen will never appear in any program run.

### **gwdtapp.ini [Screens] PstDir=<alphanum>**

*keyword; default = show*

Determines whether, or under what circumstances, the *Specify Directory for Migrated Archive* screen will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run). For example, *PstDir=skip* tells the program to display the screen only if the information it collects cannot be determined by other entries or the operating environment.

### **gwdtapp.ini [Screens] Summary=<alphanum>**

*keyword; default = show*

Determines whether, or under what circumstances, the *Selection Summary* screen will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run) — but since this screen does not collect any information, the skip and silent options are equivalent. For example, *Summary=skip* tells the program to display the *Summary* screen only if the information it collects cannot be determined by other entries or the operating environment — so this screen will never appear in any program run.

### **gwdtapp.ini [Screens] Welcome=<alphanum>**

*keyword; default = show*

Determines whether, or under what circumstances, the *Welcome* screen will appear in program runs. Valid values are *show* (screen appears in every run), *skip* (screen appears only if the information it ordinarily obtains from the user cannot be otherwise obtained or inferred), or *silent* (screen never appears in any run) — but since this screen does not collect any information, the skip and silent options are equivalent. For example, *Welcome=skip* tells the program to display the *Welcome* screen only if the information it collects cannot be determined by other entries or the operating environment — so this screen will never appear in any program run.

## **gwdtapp.ini [ServerData] section**

### **gwdtapp.ini [ServerData] MigrateJunkFolder=<#>**

*boolean; default = 1*

Determines whether the program will migrate the contents of the server junk mail folder. *MigrateJunkFolder=0* tells the program to not migrate the junk folder. By default (*MigrateJunkFolder=1*), the junk folder will be migrated.

### **gwdtapp.ini [ServerData] MigrateCalData=<#>**

*boolean; default = 1*

Determines whether the program will migrate server calendar data. *MigrateCalData=0* tells the program to not migrate server calendar data. By default (*MigrateCalData=1*), the program will migrate server calendar data.

## gwdtapp.ini [ServerData] MigrateTrashFolder=<#>

*boolean; default = 0*

Tells the program whether to migrate the server trash folder when the *Specify Data for Migration* screen does not appear, or sets the associated checkbox in that screen to be marked by default.

*MigrateTrashFolder=0* tells the program to *not* migrate the server trash folder, or sets the associated checkbox to be unmarked by default. The most common use of this parameter is to provide the necessary entry value when the program is configured to hide the *Specify Data* screen. The SSDM chapter of the *MFG Scenarios Guide* (see *Customizing the SSDM*) explains how this parameter relates to other parameters used to customize the SSDM.

Quest provides software solutions for the rapidly-changing world of enterprise IT. We help simplify the challenges caused by data explosion, cloud expansion, hybrid datacenters, security threats, and regulatory requirements. We are a global provider to 130,000 companies across 100 countries, including 95% of the Fortune 500 and 90% of the Global 1000. Since 1987, we have built a portfolio of solutions that now includes database management, data protection, identity and access management, Microsoft platform management, and unified endpoint management. With Quest, organizations spend less time on IT administration and more time on business innovation. For more information, visit [www.quest.com](http://www.quest.com).

## Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at <https://support.quest.com>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.

## A

ACLRootVisibleOnly=, 39  
ACLs=, 39, 56  
AddressListName=, 30  
AddressPolicyName=, 30  
AdminAcctPoolFirstAccountSeq=, 30  
AdminAcctPoolRoleGroupName=, 31  
AdminApiLockWaitSecs=, 42  
AdminApiRetries=, 42  
AdminDomain=, 31  
AdminPassword=, 31  
AdminProfile=, 30  
AdminUsername=, 31  
AllowAdd=, 9  
AllowSearch=, 9  
AllowUpdateOfAttributes=, 11  
AltAddressAttribute=, 11  
AlwaysExpandRecips=, 42  
AlwaysUseOldAlias=, 25  
AlwaysUseOldExchDn=, 12, 25  
AnyPst=, 56  
AppDoesArch=, 56  
AppDoesMail=, 56  
AppDoesPabs=, 56  
ArchiveDest=, 39, 57  
ArchiveDestServerArchive=, 57  
ArchiveDir=, 42  
ArchiveFolderName=, 39  
ArchiveResolveAttendees=, 31, 52  
AskDestClientType=, 56  
AskFilter=, 57  
AttachmentRetryCount=, 39, 57  
AttachSize=, 38, 55  
Attr#=, 12  
AttrFromContact=, 26  
Attribute=, 11  
AuthMechanismType=, 32, 53  
AutoReplySubject=, 42  
AutoReplyText=, 42  
auto-set parameters, 6

## B

BadSlotSkipped=, 24

BadTypeSkipped=, 24

## C

CanAddGroupMembers=, 26  
Cancel=, 23  
CanDeleteGroupMembers=, 26  
CmgExcludeFromDirSync=, 12, 26  
Code=, 49  
CommandLine=, 39, 61  
CompaniesSkipped=, 24  
CreateMailbox, 30  
CreateMailbox=, 30  
CreateMailboxDelay=, 26  
CreateUserNameFrom=, 12  
CustomAttr#=, 13, 23

## D

data types, 7  
DebugFolder=, 43, 61  
DeleteAutocompleteMessages=, 32, 53  
DestClientType=, 56  
DirSyncEnableRemoteMailbox, 32  
DoCategories=, 42, 57  
DoGALLookupRetry=, 32, 53  
DoGroupWiseAdmin=, 39  
DoInactivityTimer=, 40  
Domain#=, 50  
Domain=, 25  
DomainPath=, 42  
DoMigrateNotesOfContacts=, 40, 58  
DoPassword=, 42  
DoRmvForwarding=, 42  
DoSetForwarding=, 42  
DoTargetAdmin=, 39  
DoVisibility=, 42  
dumpaliases=, 20  
dumpNicknameAliases=, 20

## E

EmailAddressPolicyEnabled=, 33  
EnableBPOS=, 18, 40  
EnableUsers=, 13  
EndTime=, 18, 39

Error#=, 17, 18, 22, 29, 30, 52  
ErrorCnt=, 18, 39  
EwsRetryCount=, 33  
EwsRetryWait=, 33  
ExportHTML=, 43  
ExportRTF=, 43

## F

FailureLog=, 8  
FileAsFormat=, 39  
Filter#=, 38, 55  
FilteredApptCnt=, 39  
FilteredAttachCnt=, 39  
FilteredAttachSize=, 39  
FilteredMsgCnt=, 39  
FilterUser#=, 23  
FindMethod=, 11  
Finished=, 64  
FirstCalDate=, 38, 55  
FirstMailDate=, 38, 55  
Force=, 10  
ForceAddress=, 27  
ForceTargetAdmin=, 39  
ForwardingDomain=, 30, 42  
ForwardingGwDomain=, 42  
ForwardingGwPo=, 42  
ForwardingMethod=, 40  
ForwardingSMTPDomain=, 42  
ForwardLockWaitSecs=, 43  
ForwardRetries=, 43  
FwdCsvField=, 42

## G

GroupContactsContainer=, 27  
GroupsRDN=, 25  
GroupsToProvisionPath=, 20, 39  
GroupType=, 21  
GUI-set parameters, 6  
GWIALDAPPort=, 23  
GWIALDAPSearchBase=, 23  
GWIALDAPSVR=, 23  
GWIAPrefix=, 42  
GWSoapUrl=, 24, 43, 62

## H

HomeMailboxStore=, 25  
HomeServer=, 25  
Host=, 11, 17, 25

## I

IdleConnectionTimeoutSeconds=, 19, 48

InactivityTimeoutMinutes=, 40  
IncludeGwiseProxyAdrs=, 33  
Info=, 49  
InternetAddressingEnabled=, 42  
InternetDomain=, 24

## J

JUNKMAIL=, 38, 56

## K

KeyError=, 18, 20, 39

## L

LastDate=, 38, 55  
LastMailDate=, 38, 55  
LicenseKey=, 8, 18, 20, 39, 56  
ListsExported=, 24  
LoadMethod=, 25  
LocalExchangeFederationServer, 33  
LogDir=, 18, 39  
LoginAttempts=, 44  
LoginDelay=, 44  
LoginRetryWait=, 44  
LoginWait=, 44  
LogLevel=, 18, 23, 39, 58

## M

MailEnableAttempts=, 13  
MailEnableRetryWaitSeconds=, 13  
MakeDefault=, 64  
MAPIErrorsToRetry=, 34, 54  
MapiRecipientFlag=, 13, 27  
MAPIRetryWait=, 34, 54  
MaxDiscrepancyThreshold=, 41, 58  
MaxExpandedRecips=, 44  
MaxPowerShellConnections=, 19, 49  
MaxPstSize=, 40, 58  
MBoxFromContact=, 27  
MergeAddressBookFolder=, 41, 58  
MergeAddressBookList=, 41, 58  
MergeAddressBooks=, 41, 58  
MergeContactAttribWins=, 14  
MergeFindbySMTP=, 14  
MergeGrpMembership=, 27  
MergeReqImportedFrom=, 14  
MergeReqProxyAddress=, 14  
MergeX500=, 15  
MessageRetryCount=, 34, 54  
Migrate=, 39  
MigrateArchives=, 39, 59  
MigrateCalData=, 29, 49, 51, 65

MigrateDocRefs=, 44, 62  
 MigrateEmptyFolders=, 44, 62  
 MigrateEmptySharedFolders=, 44  
 MigrateFreqContacts=, 39, 59  
 MigrateHiddenMsg=, 45  
 MigrateJunkFolder=, 29, 49, 51, 65  
 MigrateMailData=, 29, 49  
 MigrateMySubjectAppointments=, 45  
 MigrateMySubjectDocs=, 45  
 MigrateMySubjectEmails=, 45  
 MigrateMySubjectNotes=, 45  
 MigrateMySubjectPhoneMsgs=, 45  
 MigrateMySubjectTasks=, 45  
 MigrateNotesAs=, 45, 62  
 MigratePAB=, 39, 59  
 MigrateServerMail=, 39, 59  
 MigrateSharedFolders=, 46, 62  
 MigrateSharedPABs=, 46, 62  
 MigrateTrashFolder=, 29, 49, 51, 66  
 MigrateUsersSample=, 20  
 MigrateWhat=, 64  
 MigrationID=, 8, 18, 20, 39, 56  
 MovedApptCnt=, 39  
 MovedApptSize=, 39  
 MovedContactCnt=, 39  
 MovedMsgCnt=, 39  
 MovedMsgSize=, 39  
 MsgIdxEnd=, 46, 63  
 MsgIdxStart=, 46, 63  
 multipleSMTPDomain =, 21

## N

NewPassword=, 42  
 NoServer=, 30

## O

O365AccountSkuld=, 35  
 O365UsageLocation=, 35  
 ObjectsProcessed=, 24  
 OtherExported=, 24

## P

PABDest=, 39, 59  
 PABDestServerArchive=, 59  
 PabsToIgnore=, 46, 63  
 parameter defaults, 7  
 parameter format and structure, 6  
 parameter value data types, 7  
 Password=, 11, 17, 25  
 PasswordMode=, 42  
 PdlDoneCnt=, 39

PdlFailCnt=, 39  
 PiggyBack=, 10  
 PiggyBackRetry=, 10  
 PiggyBackTimeout=, 10  
 PostOffice.Domain, 47  
 PostOffice=, 42  
 Profile=, 64  
 program parameters, editing, 7  
 Progress=, 64  
 PropagationDelay=, 49  
 Provider=, 18, 20, 39, 56  
 ProvisionGroups=, 39  
 ProvisionUsers=, 15  
 ProxyAccountRights=, 47, 63  
 ProxyMode=, 42  
 ProxyPassword=, 42  
 ProxyUserid=, 42  
 PSRetryAttempts=, 15, 28  
 PSRetryWait=, 15, 28  
 PstDir=, 65  
 PstDirMode=, 48  
 PstFileName=, 35  
 PstRootDir=, 48  
 PurgePABS=, 35

## R

RemigrateMode=, 36, 54  
 RemoveSrcAddresses=, 28  
 ReplaceExistingUserName=, 14  
 ResourcesExported=, 24  
 RetryCount=, 47  
 RetryDelay=, 47  
 RetryMax=, 41  
 Rights=, 10  
 RmvForwarding=, 30  
 RunName=, 18, 39

## S

scratch parameters, 7  
 SelectedProfile=, 60  
 SelectedPstDir=, 60  
 ServerMailDest=, 39, 60  
 ServerMailDestServerArchive=, 60  
 ServerMailResolveAttendees=, 36, 55  
 ServerName=, 30  
 SetAutoReply=, 42  
 SetCmgExcludeAttr=, 15, 28  
 SetDellItemsRule=, 42  
 SetDelSentItemsRule=, 48  
 SetForwarding=, 30  
 SetPurgeRule=, 42

SetRecipPolicy=, 15  
SetResourceToAutoAccept=, 36  
SetUserAccountControl=, 16  
ShowBPOSIDAttrs=, 18  
ShowFirst=, 64  
ShowResolveAttendeesOpt=, 36  
Silent=, 41  
SkipCountByFolder=, 48, 63  
SkipIncomingSharedFolder=, 41  
SkippedMsgCnt=, 39  
SMTPTranslationTable=, 20, 39, 56  
SqlAuthMethod=, 11  
SqlConnect=, 16  
SqlPassword=, 11  
SqlServer=, 11  
SqlUserName=, 11  
StartTime=, 18, 39  
Status=, 18, 24, 39  
SuccessLog=, 8  
Summary=, 65  
SyncObjRequired=, 16

## T

TargetAliasFormatTypes, 20  
Text=, 49  
ThreadCnt=, 39  
Total=, 24  
TotalApptCnt=, 39  
TotalContactCnt=, 39  
TotalMsgCnt=, 39  
transient “scratch” parameters, 7

## U

UnknownsSkipped=, 24  
UpdateAddressList=, 36  
UpdateEmailAddressPolicy=, 36  
UseAdminAPI=, 21  
UseApiGateway=, 42  
UseLargePsts=, 39, 61  
UseMapiRecipientFlag=, 16, 28  
UsePseudoRoot=, 29  
User=, 9  
UserAccountControl=, 17  
UserContainer=, 17  
UserDoneCnt=, 18, 39  
UserErrList=, 39  
UserFailCnt=, 18, 39  
UserGroupWiseAddr=, 9  
Userid=, 17, 25  
UserList=, 19, 39  
UserListHasSearchKey=, 18

UsersExported=, 24  
UsersToMergePath=, 20  
UseSeparatePSTS=, 41  
UseTrustedAPI=, 48

## V

ValidateForwardingAddresses=, 37  
VerifyO365AdminRights=, 37  
VisibilityMode=, 42

## W

WarningCnt=, 39  
Welcome=, 65  
WorkerLimit=, 22  
WriteMessageRetryWait=, 37  
WriteMessageTryCount=, 37