




ibatch Users Guide

July 2002



© 2002 by Atrove Systems. All rights reserved.

No part of this document may be produced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Atrove Systems.

Specifications and information in this document is current or planned as of the date of publication. It is subject to change without notice.

Atrove Systems trademarks and registered trademarks

Atrove Systems™

Atrove™

InfoTrove™

ibatch™

Trademark Credits

Microsoft is a U.S. registered trademark and Windows™ and Internet Explorer™ are trademarks of Microsoft Corporation.

MacOS is a registered trademark of Apple Computer, Inc.

Sun Solaris™ is a trademark of Sun Microsystems, Inc.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

Other products mentioned may be trademarks or registered trademarks of their registered companies. All other trademarks belong to their respective owners.

Last Update: July 2002

Contents

Start Here	1
Welcome to ibatch	1
Audience	1
About this Manual	1
Getting Help.....	2
System Requirements	2
Installation Instructions	2
Downloading ibatch	3
Setting up a Path.....	3
Testing the Installation	3
Setting up the ibatch Configuration File	4
Configuration File Location.....	5
Specifying a Configuration File.....	5
Changing Your Configuration Details	5
Enabling ibatch Download	6
Introducing ibatch	7
Features.....	7
ibatch docs	7
ibatch user.....	8
ibatch contact.....	8
ibatch group	9
ibatch docs	10
Overview.....	10
Command Summary	10
docs add_doc	11
docs add_folder.....	12
docs copy	12
docs checkout.....	13
docs checkin.....	13
docs list	15

docs view	15
docs edit.....	16
ibatch user	18
Overview.....	18
Command Summary	18
user list.....	19
user view	19
user edit.....	20
user add.....	21
ibatch contact	23
Overview.....	23
Command Summary	23
contact list.....	24
contact view	24
contact edit.....	25
ibatch group.....	27
Overview.....	27
Command Summary	27
group list	28
group view	28
group edit.....	29
group add	30
Command Reference	31
Introduction.....	31
Running ibatch.....	31
ibatch Command Syntax	31
Option Syntax.....	32
Command Argument Syntax	32
Case Sensitivity	32
Quoting Command Arguments	32
Allowed Characters	33
Object Indexing	33
Indexes are Case Sensitive	33
Objects Path Delimiter	33
Wildcards.....	34
Document Number in Filename.....	34
Standard In, Standard Out, Standard Error	35
Standard Input	35

Standard Output.....	36
Standard Error	36
Meta Data.....	36
Entering Meta Data Interactively	36
Selecting from Meta Data Choices	37
Default Meta Data Value.....	37
Entering Multi-Line Data Interactively	37
Entering Meta Data via File Input.....	37
File Meta Data Syntax	38
File Meta Data Keywords	38
Applying Meta Data to Multiple Command Arguments	39
Entering Multi-Line File Data	39
Return codes	40
Permissions and Security.....	40
Mail Considerations for Bulk Operations.....	41
Object Locks	41

IBATCH USERS GUIDE

START HERE

Welcome to ibatch

The ibatch program provides a powerful command line interface to Atrove's InfoTrove document management system. It is the ideal tool for performing bulk operations or for automating library interactions.

Audience

Users of ibatch will find it helpful to have some understanding of batch and scripting concepts, and to be familiar with operating system shells.

About this Manual

This user manual is organized into five parts that will help you become familiar with ibatch features and common batch processing tasks. Examples are provided throughout to help you learn basic features and concepts.

“Introducing ibatch” gives a brief introduction to the ibatch program and describes key features.

The “ibatch docs” section describes the ibatch docs command and provides usage examples of the various ibatch docs operations.

The “ibatch user” section describes the ibatch user command and provides usage examples of the various ibatch user operations.

The “ibatch contact” section describes the ibatch contact command and provides usage examples of the various ibatch contact operations.

The “ibatch group” section describes the ibatch group command and provides usage examples of the various ibatch group operations.

“Command Reference” is reference guide for the ibatch command. It covers topics such as command syntax, working with files, meta data usage, and the use of standard input and output.

Getting Help

Help is available via the following methods:

Command Line Help The **ibatch** program has in-built context sensitive help accessible via the **-h** command line option.

Web Help The latest versions of all Atrove user manuals and installation guides are available via Atrove Systems web site at <http://www.atrove.com>.

Technical Support is available by email or telephone to users covered under warranty or a Service Agreement. We also offer e-mail only support for evaluation users. Email and telephone contact information is available from the Atrove Systems web site under the Contact Details section <http://www.atrove.com/about/contact.htm>.

System Requirements

The **ibatch** command line program is distributed for a number of operating systems including Windows 95/98/Me/NT/2000, Sun Solaris 2.4 or later, and Redhat 6.0 or later. If you require support for other operating systems please email Atrove Systems and we can provide a version for your specific system.

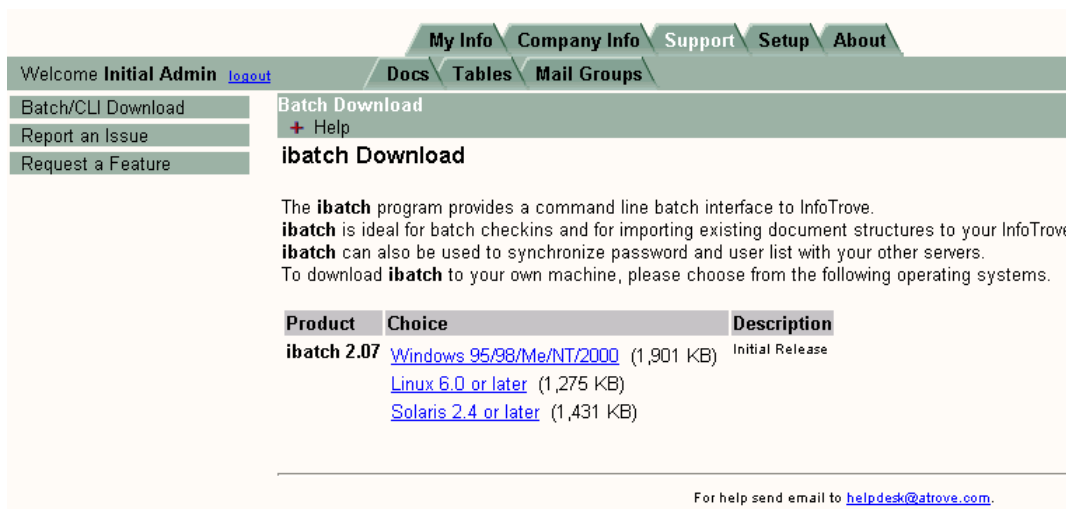
To run **ibatch** on your local computer, you will need approximately 2Mbytes of free disk space, and 20Mbytes of virtual memory.

Installation Instructions

The **ibatch** command line program is included in the InfoTrove distribution. It is available for download under the InfoTrove **Support** application tab. By default, only admin users are enabled to see Support applications. If you can not see the Support application tab, and wish to use **ibatch**, ask your administrator to enable access.

DOWNLOADING IBATCH

To download ibatch to your computer, log onto your InfoTrove server, and click on the **Support** application tab. The Display Frame changes to show the Batch/CLI Download page.



The screenshot shows the InfoTrove web interface. At the top, there are navigation tabs: My Info, Company Info, Support, Setup, and About. Below these, there are sub-tabs: Docs, Tables, and Mail Groups. The main content area is titled 'Batch Download' and includes a '+ Help' link. The primary heading is 'ibatch Download'. The text describes the program as a command line interface and provides instructions on how to download it for different operating systems. A table lists the available download choices for 'ibatch 2.07'.

Product	Choice	Description
ibatch 2.07	Windows 95/98/Me/NT/2000	(1,901 KB) Initial Release
	Linux 6.0 or later	(1,275 KB)
	Solaris 2.4 or later	(1,431 KB)

For help send email to helpdesk@atrove.com.

Next click on the blue link corresponding to your particular operating system. Depending on your browser, a form may appear asking if you want to run or save the program. Select save and then browse to select a destination folder on your local machine where you wish to save the ibatch program.

SETTING UP A PATH

You can run ibatch directly from the folder where you saved it, or you can set up a path to this folder, enabling you to run ibatch from anywhere. For example on Windows 2000, the path can be set via:

Start->Settings->Control Panel->System->Advanced->Environment Variables->System Variables->Edit.

For other operating systems, please consult your system documentation for information on setting the path variable.

TESTING THE INSTALLATION

To run ibatch, you need to open up a shell window or command prompt window on your machine.

For example, on Windows 2000, select **Run** from the **Start** menu and type **cmd** and click OK. In Windows 95/98/Me/NT follow the same procedure but type **command** instead of **cmd**.

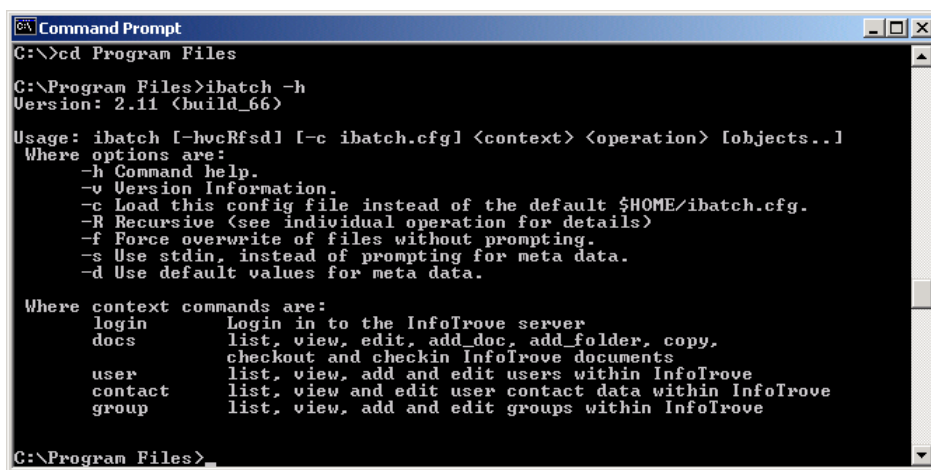
A command prompt window will appear. Next, if you have not set up your operating system **PATH** variable, change directory to the folder where you stored ibatch. For example, if you stored ibatch under **c:\Program Files**, then you would enter:

```
C:  
cd \Program Files
```

Finally, to test the ibatch installation, type

```
ibatch -h
```

at the command prompt. The help message should appear as shown below.



```
Command Prompt  
C:\>cd Program Files  
C:\Program Files>ibatch -h  
Version: 2.11 (build_66)  
  
Usage: ibatch [-hvcRfsd] [-c ibatch.cfg] <context> <operation> [objects..]  
Where options are:  
-h Command help.  
-v Version Information.  
-c Load this config file instead of the default $HOME/ibatch.cfg.  
-R Recursive (see individual operation for details)  
-f Force overwrite of files without prompting.  
-s Use stdin, instead of prompting for meta data.  
-d Use default values for meta data.  
  
Where context commands are:  
login      Login in to the InfoTrove server  
docs       list, view, edit, add_doc, add_folder, copy,  
           checkout and checkin InfoTrove documents  
user       list, view, add and edit users within InfoTrove  
contact    list, view and edit user contact data within InfoTrove  
group      list, view, add and edit groups within InfoTrove  
C:\Program Files>
```


SETTING UP THE IBATCH CONFIGURATION FILE

When you first run ibatch, the program will prompt you for your InfoTrove server address, your username, and your password. This information is encrypted and stored in a local configuration file, **ibatch.cfg**, to prevent you having to re-enter it every time you run ibatch. For example:

```
C:\>ibatch docs list  
Enter The InfoTrove Server URL Address [http://my_server:5000]: http://my_server:5000  
Enter Your User Name [admin]: jbloggs  
Enter Your Password [*****]: my_password  
/Docs/
```

When prompted for the Server URL, enter the web address of your InfoTrove server - just as you would if you were using InfoTrove.

The Server URL format can be any one of the following:



START HERE

Installation Instructions

- Server Name (e.g. `http://myInfoTroveserver` or `http://myInfoTroveserver.mycompany.com`)
- Server Name and Port Number (e.g. `http://myInfoTroveserver:5000`)
- Server IP Address (e.g. `http://192.111.65.55`)

Configuration File Location

By default, the configuration file **ibatch.cfg** is stored in your home area defined by the environment variable **HOME**.

On Windows based systems, the **HOME** variable is often not defined, in which case the configuration file search will default to **c:**.

If you are on a Windows based system, you can accept the **c:** default, or you can choose to define the **HOME** variable as described in your system documentation.

For example on Windows 2000, the **HOME** variable can be set via:

Start->Settings->Control Panel->System->Advanced->Environment Variables->User Variables->Edit

For other operating systems, please consult your system documentation for information on setting the **HOME** variable.

Specifying a Configuration File

You can specify a configuration file, rather than use the default configuration file, by using the **-c** option. For example:

```
ibatch docs list -c "C:/My Documents/new_config.txt"
```

Changing Your Configuration Details


You can change your configuration details at anytime by using the **login** command to prompt for new details. For example:

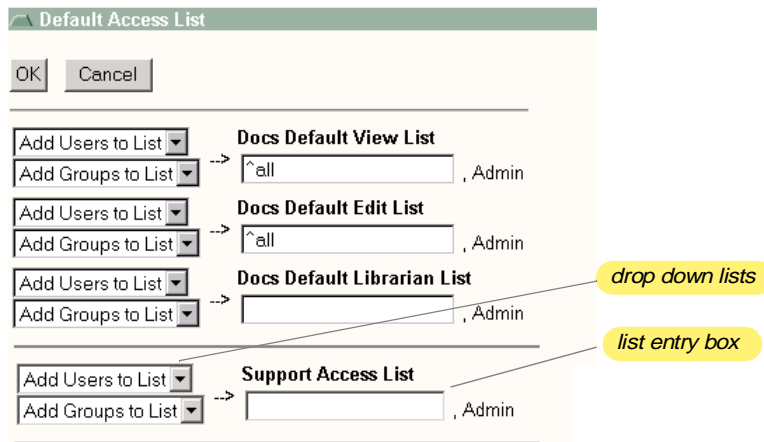
```
ibatch login
```

Your new details will be saved in the default configuration file.

ENABLING IBATCH DOWNLOAD

The ibatch command line program can be downloaded by all users with support access privileges.

By default, InfoTrove ships with only Admin users set up for support access. The default setting can be modified under the **SetUp->Default Access List** menu. To change the default access settings click on the  Edit pen to bring up the Default Access List form.



The screenshot shows the 'Default Access List' dialog box. It has 'OK' and 'Cancel' buttons at the top left. Below are four sections, each with 'Add Users to List' and 'Add Groups to List' dropdown menus and a text entry box. The first three sections are for 'Docs' (View, Edit, Librarian) and the last is for 'Support Access'. The 'Docs Default View List' and 'Docs Default Edit List' entry boxes contain '^all'. The 'Docs Default Librarian List' and 'Support Access List' entry boxes are empty. Two yellow callout boxes with arrows point to the dropdown menus and the text entry boxes, labeled 'drop down lists' and 'list entry box' respectively.

Add users to the Support Access List by using the drop down menus, or by typing directly in the list entry boxes as described in the InfoTrove Users Manual. When you have finished click OK to accept the changes.

INTRODUCING IBATCH

The ibatch program provides a powerful command line interface to Atrove's InfoTrove document management system. It allows users to construct batch files for performing functions such as adding bulk documents to create an initial library, for automatically creating an entire reference copy of a library or folder, or for performing periodic tasks such as the synchronization of usernames and passwords between servers.

Additionally ibatch can be used to facilitate interworking between InfoTrove and other third party information management systems.

The following section provides detailed information about the key features of ibatch.

Features

There are four main ibatch command contexts - docs, user, contact and group. Each command context matches a major InfoTrove application or function.

ibatch docs provides a set of operations for performing standard InfoTrove document functions such as adding new documents and checking in and out documents.

ibatch user provides a set of operations for adding users to InfoTrove, and for viewing and modifying user profile details.

ibatch contact provides a set of operations for viewing and changing user contact details.

ibatch group provides a set of operations for adding groups to InfoTrove, and for viewing and modifying group profile details.

ibatch docs

The ibatch docs command supports the following operations.

- **docs add_doc** to add a new document or set of documents to InfoTrove.
- **docs add_folder** to create a new folder or set of folders in InfoTrove.
- **docs copy** copies an InfoTrove file or folder to your local disk.
- **docs checkout** checks out an InfoTrove file or folder to your local disk and reserves the document for editing.
- **docs checkin** returns an edited document to the library and removes the edit lock.
- **docs list** all the files in the specified InfoTrove folder.

- **docs view** shows the meta data properties of the specified InfoTrove file or folder.
- **docs edit** modifies the meta data properties of the specified InfoTrove file or folder.

For a usage description and examples of the various docs command operations refer to the [ibatch docs](#) section.

ibatch user

The `ibatch user` command supports the following operations.

- **user list** produces a list all InfoTrove users.
- **user view** shows the specified user/s profile information.
- **user edit** modifies the specified user/s profile information.
- **user add** creates an InfoTrove account for the specified user.

For a usage description and examples of the various user command operations refer to the [ibatch user](#) section.

ibatch contact

The `ibatch contact` command supports the following operations.

- **contact list** produces a list all InfoTrove users.
- **contact view** shows the specified user/s contact information.
- **contact edit** modifies the specified user/s contact information.

For a usage description and examples of the various contact command operations refer to the [ibatch contact](#) section.

ibatch group

The `ibatch group` command supports the following operations.

- **group list** produces a list all InfoTrove groups.
- **group view** shows the specified group/s profile information.
- **group edit** modifies the specified group/s profile information.
- **group add** creates a new InfoTrove group.

For a usage description and examples of the various group command operations refer to the [ibatch group](#) section.

IBATCH DOCS

Overview

ibatch docs provides a set of operations for performing standard InfoTrove document functions such as adding new documents and checking in and out documents.

This section provides a brief usage description and examples for each of the **ibatch docs** operations. For a full description of the **ibatch** command syntax refer to the [Command Reference](#) section.

Note that in the examples below, all the file and folder arguments are shown quoted with “double quotes”. The quotes are required by some shells to protect special non-alphanumeric characters. If you don’t use these characters in your file and folder names, the quotes may not be necessary.

Command Summary

ibatch docs can be run by typing **ibatch docs** at the command prompt followed by the desired docs operation.

The **ibatch docs** command syntax is:

```
ibatch docs <operation> [-options] [files...] [folder...]
```

A list of available **<operations>** is available using the **-h** help option.

```
C:\>ibatch docs -h

Usage: ibatch docs <operation> [-hRfsd] [files...] [folder...]
Where operation commands are:
list           List all documents
view          View object meta data
edit          Edit object meta data
add_doc       Add a new Document
add_folder    Add a new Folder
copy          Copy InfoTrove files to local disk
checkout      Checkout and lock InfoTrove files to local disk
checkin       Checkin local file to InfoTrove
```

The **[file...]** and **[folder...]** command arguments, and the command **[-options]**, depend on the individual operation. The arguments and options for each operation are detailed below.

docs add_doc

The **docs add_doc** command adds the local file/s to the specified InfoTrove folder. It can be used to add anything from a single new document, to an entire folder tree of documents. It is an ideal command for initially populating your InfoTrove library with your existing legacy documents.

```
Usage: ibatch docs add_doc [-Rsd] <local_file|local_folder>... <InfoTrove_folder>
Where options are:
-R Recursive submit of all files and folders below <local_folder>
-s Use the data on STDIN to edit the meta data.
-d Use default values for meta data.
```

Examples:

To add a single file called *Local.doc* to the InfoTrove common area:

```
ibatch docs add_doc "C:/My Documents/Local.doc" "/Docs/Common"
```

To add all files in the local folder use the wildcard *:

```
ibatch docs add_doc "C:/My Documents/*" "/Docs/Common"
```

To add all files and sub-folders use the **-R** recursive option. This option will recreate the local folder structure within InfoTrove:

```
ibatch docs add_doc "C:/My Documents/*" "/Docs/Common" -R
```

Usage Tips:

- The **add_doc** command by default will interactively prompt the user for meta data corresponding to the destination InfoTrove folder (e.g.author, abstract).
- When adding bulk documents, it may be preferable to add the meta data automatically. As an alternative to manual entry of meta data, one of two automatic options can be specified. The **-d** option can be specified to automatically accept the default values for the specified destination folder/s. Or the **-s** option can be specified to accept meta data values from standard input such as a file (see [Standard Input](#)). If both options are inadvertently specified the **-d** option will take precedence.
- The revision of new documents will default to either **0.1** or **1.0** depending on the properties of the InfoTrove folder the documents are being submitted to.

docs add_folder

The **docs add_folder** command creates the specified folder within InfoTrove. It can be used to create a single folder, or a folder structure.

```
Usage: ibatch docs add_folder <InfoTrove_folder>...
```

Examples:

To create a single folder called `New_folder` in the InfoTrove `/Docs/Common` area:

```
ibatch docs add_folder "Docs/Common/New_folder"
```

To create a new folder tree `Marketing/Brochures/ProductX` in The InfoTrove `/Docs` area:

```
ibatch docs add_folder "Docs/Marketing/Brochures/ProductX"
```

docs copy

The **docs copy** command copies an InfoTrove file or folder to your local disk. It can be used to copy a single file or a group of files. It is an ideal command for making a complete reference copy of a folder or of the entire InfoTrove library.

```
Usage: ibatch docs copy [-Rf] <InfoTrove_file|InfoTrove_folder>... <local_folder>  
Where options are:  
-R Recursive copy of all files and folders below <InfoTrove_folder>  
-f Force overwrite of files without prompting.
```

Examples:

To copy a single file called `test.doc` from the InfoTrove `/Docs/Common` area to your local disk:

```
ibatch docs copy "/Docs/Common/test.doc" "C:/My Documents"
```

To copy all the files from the InfoTrove `/Docs/Common` area to your local disk:

```
ibatch docs copy "/Docs/Common/*" "C:/My Documents"
```

To copy all the files and sub-folders from the InfoTrove `/Docs/Common` area to your local disk:

```
ibatch docs copy "/Docs/Common/" "C:/My Documents" -R
```

The above command will copy everything within Common and below to your local disk, including the Common folder itself.

The following will achieve the same thing, except the Common folder itself will not be copied - just its contents.

```
ibatch docs copy "/Docs/Common/*" "C:/My Documents" -R
```

Usage Tips:

- If a you attempt to copy a file out of InfoTrove that already exists on your local disk, ibatch will ask if you want to overwrite the local file. If you want to turn off the prompt, and automatically overwrite, use the **-f** option.

docs checkout

The **docs checkout** command checks out an InfoTrove file or folder to your local disk. It can be used to checkout a single file or a group of files.

```
Usage: ibatch docs checkout [-Rf] <InfoTrove_file|InfoTrove_folder>... <local_folder>  
Where options are:  
-R Recursive checkout of all files below <InfoTrove_folder>  
-f Force overwrite of files without prompting.
```

Examples:

To checkout a single file called *test.doc* from the InfoTrove */Docs/Common* area to your local disk:

```
ibatch docs checkout "/Docs/Common/test.doc" "C:/My Documents"
```

To checkout all the files from the InfoTrove */Docs/Common* area to your local disk:

```
ibatch docs checkout "/Docs/Common/*" "C:/My Documents"
```

To checkout the contents of the */Docs/Common* folder and all its files and sub-folders to your local disk:

```
ibatch docs checkout "/Docs/Common/" "C:/My Documents" -R
```

docs checkin

The **docs checkin** command checks in a local file to an InfoTrove file or folder. It can be used to checkin a single file or a group of files.

The checkin command includes a special search capability. If no target InfoTrove folder is specified, checkin will automatically attempt to find the file within the InfoTrove library.

```
Usage: ibatch docs checkin [-Rsd] <local_file>... [InfoTrove_file|InfoTrove_folder]
-R Recursive checkin of all files below <InfoTrove_folder>
-s Use the data on STDIN to edit the meta data.
-d Use default values for meta data.
```

Examples:

To check back in a single file called *test.doc* by specifying the full InfoTrove file path corresponding to where the original file was checked out from:

```
ibatch docs checkin "C:/My Documents/test.doc" "/Docs/Common/test.doc" or
ibatch docs checkin "C:/My Documents/test.doc" "/Docs/Common"
```

To checkin a single file called *test.doc* using the search capability:

```
ibatch docs checkin "C:/My Documents/test.doc"
```

To checkin just the files in *C:/My_Documents* to the folder */Docs/Common*:

```
ibatch docs checkin "C:/My Documents/*" "/Docs/Common"
```

To check back in an entire folder tree including all files and sub-folders to */Docs/Common*:

```
ibatch docs checkin "C:/My Documents/Common" "/Docs" -R
```

where the tree was previously checked out using:

```
ibatch docs checkout "/Docs/Common/" "C:/My Documents" -R
```

Usage Tips:

- Checkin will fail if the document has not been previously checked out for edit.
- When checking in a document using the search capability, the document must be unique within the specified InfoTrove folder. If no InfoTrove folder is specified, the document must be unique within the entire InfoTrove library.
- The **checkin** command by default will interactively prompt the user for meta data (e.g. the new Revision and the Change Log).
- When checking in bulk documents, it may be preferable to add the meta data automatically. As an alternative to manual entry of meta data, one of two automatic options can be specified. The **-d** option can be specified to automatically accept the default values for the specified destination folder/s. Or the **-s** option can be specified to accept meta data values from standard input such as

a file (see [Standard Input](#)). If both options are inadvertently specified the **-d** option will take precedence.

docs list

The **docs list** command lists all the files in the specified InfoTrove folder.

```
Usage: ibatch docs list [-R] [folder]
Where options are:
-R Recursive listing of all files and folders below [InfoTrove_folder]
```

Examples:

To list all the files in the InfoTrove /Docs/Common folder:

```
ibatch docs list "/Docs/Common"
```

To list all the files and sub-folders in and below the InfoTrove /Docs/Common folder:

```
ibatch docs list "/Docs/Common" -R
```

To list all the files in the entire InfoTrove library:

```
ibatch docs list "/Docs" -R
```

docs view

The **docs view** command views the associated meta data of the specified InfoTrove file or folder. It shows the same information as displayed on the corresponding InfoTrove status page. For example, for files it displays meta data including Title, Author, Revision, Filename, Filetype and the various Access List settings.

```
Usage: ibatch docs view [-R] <InfoTrove_file|InfoTrove_folder>...
Where options are:
-R Recursive view of all files and folders below <InfoTrove_folder>
```

Examples:

To view the meta data for the InfoTrove /Docs/Common folder:

```
ibatch docs view "/Docs/Common"
```

To view the meta data for the file *test.doc* in the InfoTrove /Docs/Common folder:

ibatch docs view "/Docs/Common/test.doc"

To view the meta data for all the files and sub-folders in and below the InfoTrove /Docs/Common folder:

ibatch docs view "/Docs/Common" -R

To view the meta data for all the files and folders in the entire InfoTrove library:

ibatch docs view "/Docs" -R

docs edit

The **docs edit** command modifies the meta data properties of the specified InfoTrove file or folder. It can be used to change the meta data properties on a single file, or change a certain property on all the files in the library.

```
Usage: ibatch docs edit [-Rs] <InfoTrove_file|InfoTrove_folder>...
Where options are:
-R Recursive edit of all files and folders below <InfoTrove_folder>
-s Use the data on STDIN to edit the meta data.
```

Examples:

To edit the meta data for the InfoTrove /Docs/Common folder:

ibatch docs edit "/Docs/Common"

To edit the meta data for the file *test.doc* in the InfoTrove /Docs/Common folder:

ibatch docs edit "/Docs/Common/test.doc"

To edit the meta data interactively for all the files and sub-folders in and below the InfoTrove /Docs/Common folder:

ibatch docs edit "/Docs/Common" -R

To edit the meta data for the above example using an input file called *meta_data.txt* to specify the meta data values:

ibatch docs edit "/Docs/Common" -R -s < meta_data.txt

For detailed information on using file input, see [Entering Meta Data via File Input](#).

Usage Tips:

- The **edit** command by default will interactively prompt the user for meta data corresponding to the destination InfoTrove file or folder (e.g.author, abstract).



IBATCH DOCS

docs edit

- When changing the meta data for a large number of files, it may be more convenient to use a file to input the meta data changes. The **-s** option can be specified to accept meta data values from a file (see [Standard Input](#)).

IBATCH USER

Overview

ibatch users provides a set of operations for adding users to InfoTrove, and for viewing and modifying user profile details.

This section provides a brief usage description and examples for each of the **ibatch user** operations. For a full description of the **ibatch** command syntax refer to the [Command Reference](#) section.

Note that in the examples below, the user name argument is shown quoted with “double quotes”. The quotes are required by some shells to protect special non-alphanumeric characters. If you don’t use these characters in your user names, the quotes may not be necessary.

Command Summary

ibatch user can be run by typing **ibatch user** at the command prompt followed by the desired user operation.

The **ibatch user** command syntax is:

```
ibatch user <operation> [-options] [user_name...]
```

A list of available **<operations>** is available using the **-h** help option.

```
C:\>ibatch user -h

Usage: ibatch user <operation> [-hsd] [user_name...]
Where operation commands are:
  list           List users
  view           View user data
  edit           Edit user data
  add            Add new user
```

The command **[-options]**, depend on the individual operation. The options for each operation are detailed below.

user list

The **user list** command produces a list all InfoTrove users. Once the user list is known, user details can be queried or modified using the view and edit commands.

```
Usage: ibatch user list [user_name...]
```

Examples:

To list all users in InfoTrove:

```
ibatch user list
```

user view

The **user view** command views the specified user/s profile information. It shows the same information as displayed on the InfoTrove **My Info->My Profile** page.

```
Usage: ibatch user view <user_name>...
```

Examples:

To view the profile information for all users:

```
ibatch user view *
```

To view the profile information for the user jbloggs:

```
ibatch user view "jbloggs"
```

```
C:\> ibatch user view "jbloggs"
----- jbloggs -----
ID                               ='3'
FULL_NAME                       ='Joe J. Bloggs'
USER_NAME                       ='jbloggs'
PERMISSION                      ='Admin'
PASSWD                          ='BEyJQX4gJdb1E'
NEW_DOCUMENT                    ='Mail Me'
DOCUMENT_CHECKIN                ='Mail If View Access'
EMAIL                           ='jbloggs@atrove.com'
GROUPS                          ='^all'
```

user edit

The **user edit** command modifies the specified user/s profile information. It can be used to change the profile information for a single user, or to change a particular property for all users in the library.

```
Usage: ibatch user edit [-s] <user_name>...  
Where options are:  
-s Use the data on STDIN to edit the meta data.
```

Examples:

To edit the profile information for user jbloggs:

```
ibatch user edit "jbloggs"
```

```
C:\>ibatch user edit "jbloggs"  
----- edit jbloggs -----  
Enter the FULL_NAME [Joe]: Joe J. Bloggs  
Enter the Encrypted PASSWD [BEyJQX4gJdb1E]:  
  1 - Mail Me _____ List of Choices  
  2 - No Mail  
Enter the NEW_DOCUMENT number (No Mail) [2]: 1 _____ Enter number for Choice  
  1 - Mail If Read Before  
  2 - Mail If Edit Access  
  3 - Mail If View Access  
  4 - No Mail  
Enter the DOCUMENT_CHECKIN number (No Mail) [4]: 3  
Enter the EMAIL [jbloggs@atrove.com]: _____ Leave blank to accept default  
  
C:\>
```

Usage Tips:

- By default, the **edit** command will interactively prompt for input meta data. Where the data to be entered is selectable from a pre-defined list, the list choices are numbered allowing the choice to be conveniently selected.
- When being prompted for meta data, a default value is shown in square brackets []. To accept the default hit enter or return.

IBATCH USER

user add

- It is possible to change a particular property for all users in the library (e.g. mail options), by using the **-s** option to read the information from a file. For more information on using file input refer to [Entering Meta Data via File Input](#).
- The password is shown in encrypted form consistent with the operating system hosting the InfoTrove server. Generally this field should not be edited via the interactive method. It is included primarily to facilitate automatic batch synchronization with your operating system password file.

user add

The **user add** command creates an InfoTrove account for the specified user. It can be run interactively to create a single user account, or can be used with file input to create a number of users. It is an ideal command for initially adding users to InfoTrove.

```
Usage: ibatch user add [-sd] <user_name>...
Where options are:
-s Use the data on STDIN to edit the meta data.
-d Use default values for meta data.
```

Examples:

To add a single user called "mary":

ibatch user add "mary"

```
C:\>ibatch user add "mary"
----- add mary -----
Enter the FULL_NAME [Joe Bloggs]: Mary Citizen
  1 - Admin
  2 - Normal _____ List of Choices
  3 - View Only
Enter the PERMISSION number (Normal) [2]: 1 Enter number for Choice
Enter the Encrypted PASSWD [fV6Dcb8VGcpyl]:
  1 - Mail Me
  2 - No Mail
Enter the NEW_DOCUMENT number (Mail Me) [1]:
  1 - Mail If Read Before
  2 - Mail If Edit Access
  3 - Mail If View Access
  4 - No Mail
Enter the DOCUMENT_CHECKIN number (Mail If View Access) [3]:
Enter the EMAIL []: mary@atrove.com
Enter the GROUPS [^all]: _____ Leave blank to accept default

C:\>
```

Usage Tips:

- By default, the **add user** command will interactively prompt for input meta data. Where the data to be entered is selectable from a pre-defined list, the list choices are numbered allowing the choice to be conveniently selected.
- When being prompted for meta data, a default value is shown in square brackets []. To accept the default hit enter or return.
- The **-s** option can be used to read the meta data information from a file. Using this option, a user script can be created to automatically add users from an existing operating system password file. For more information on using file input refer to [Entering Meta Data via File Input](#).

IBATCH CONTACT

Overview

ibatch contact provides a set of operations for viewing and changing user contact details.

It provides control over the same information contained on the InfoTrove **My Info->My Contact Info** page.

This section provides a brief usage description and examples for each of the **ibatch contact** operations. For a full description of the **ibatch** command syntax refer to the [Command Reference](#) section.

Note that in the examples below, the user name argument is shown quoted with “double quotes”. The quotes are required by some shells to protect special non-alphanumeric characters. If you don’t use these characters in your user names, the quotes may not be necessary.

Command Summary

ibatch contact can be run by typing **ibatch contact** at the command prompt followed by the desired contact operation.

The **ibatch contact** command syntax is:

```
ibatch contact <operation> [-options] [user_name...]
```

A list of available **<operations>** is available using the **-h** help option.

```
C:\>ibatch user -h

Usage: ibatch contact <operation> [-h] [user_name...]
Where operation commands are:
  list          List users
  view         View user contact data
  edit         Edit user contact data
```

The command **[-options]**, depend on the individual operation. The options for each operation are detailed below.

contact list

The **list** command produces a list all InfoTrove contact users. Once the user list is known, user contact details can be queried or modified using the view and edit commands.

```
Usage: ibatch contact list [user_name...]
```

Examples:

To list all users in InfoTrove:

```
ibatch contact list
```

contact view

The **contact view** command views the specified user/s contact information. It shows the same information as displayed on the InfoTrove **My Info->My Contact Info** page.

```
Usage: ibatch contact view <user_name>...
```

Examples:

To view the contact information for all users:

```
ibatch contact view *
```

To view the contact information for the user jbloggs:

IBATCH CONTACT

contact edit

ibatch contact view "jbloggs"

```
C:\>ibatch contact view "jbloggs"
----- jbloggs -----
ID                               ='10'
USER_NAME                        ='jbloggs'
EMAIL                            ='jbloggs@atrove.com'
WORK_PHONE                       ='(408) 545-7732'
HOME_PHONE                       ='(408) 631-5444'
FAX                              ='(408) 545-7700'
MOBILE/CELL                      ='(510) 504-0491'
PAGER                           ='(800) 756-8888'
ADDRESS                          ='2130 North First Street
'
CITY                             ='San Jose'
STATE                            ='CA'
POSTCODE                         ='95131'
COUNTRY                          ='USA'

C:\>
```

contact edit

The **contact edit** command modifies the specified user/s contact information. It can be used to change the contact information for a single user, or to change a particular property for all users in the library.

```
Usage: ibatch contact edit [-sd] <user_name>...
Where options are:
-s Use the data on STDIN to edit the meta data.
```

Examples:

To edit the contact information for user jbloggs:

ibatch contact edit "jbloggs"

```
C:\>ibatch contact edit "jbloggs"
----- edit jbloggs -----
Enter the EMAIL [jbloggs@atrove.com]:
Enter the WORK_PHONE [(408) 545-7732]: (408) 545-7733
Enter the HOME_PHONE [(408) 631-5444]:
Enter the FAX [(408) 545-7700]:
Enter the MOBILE/CELL [(510) 504-0491]:
Enter the PAGER [(800) 756-8888]:
----- Original Text -----
2130 North First Street

-----
Enter the ADDRESS [Enter a single . on a new line to finish]
[Or enter <CR> to keep the original text]

Enter the CITY [San Jose]:
Enter the STATE [CA]:
Enter the POSTCODE [95131]:
Enter the COUNTRY [USA]:

C:\>
```

Usage Tips:

- By default, the **contact edit** command will interactively prompt for input meta data. When being prompted for meta data, a default value is shown in square brackets []. To accept the default hit enter or return.
- When entering the address field, which typically spans several lines, a single period "." on a new line is used to denote the end of the address entry.
- It is possible to change a particular property for all users in the library (e.g. company address), by using the **-s** option to read the information from a file. For more information on using file input refer to [Entering Meta Data via File Input](#).

IBATCH GROUP

Overview

ibatch group provides a set of operations for adding groups to InfoTrove, and for viewing and modifying group profile details.

This section provides a brief usage description and examples for each of the **ibatch group** operations. For a full description of the **ibatch** command syntax refer to the [Command Reference](#) section.

Note that in the examples below, the group name argument is shown quoted with “double quotes”. The quotes are required by some shells to protect special non-alphanumeric characters. If you don’t use these characters in your group names, the quotes may not be necessary.

Command Summary

ibatch group can be run by typing **ibatch group** at the command prompt followed by the desired group operation.

The **ibatch group** command syntax is:

```
ibatch group <operation> [-options] [group_name...]
```

A list of available **<operations>** is available using the **-h** help option.

```
C:\>ibatch group -h

Usage: ibatch group <operation> [-hsd] [group_name...]
Where operation commands are:
  list           List groups
  view          View group data
  edit          Edit group data
  add           Add new group
```

The command **[-options]**, depend on the individual operation. The options for each operation are detailed below.

group list

The **group list** command produces a list all InfoTrove groups. Once the group list is known, group details can be queried or modified using the view and edit commands.

```
Usage: ibatch group list [group_name...]
```

Examples:

To list all groups in InfoTrove:

```
ibatch group list
```

group view

The **group view** command views the specified group/s profile information.

```
Usage: ibatch group view <group_name>...
```

Examples:

To view the profile information for all groups:

```
ibatch group view *
```

To view the profile information for the group ^sales:

```
ibatch group view "^sales"
```

```
C:\> ibatch group view "^sales"  
----- ^sales -----  
ID                                     ='7'  
GROUP_NAME                             ='^sales'
```

group edit

The **group edit** command modifies the specified group/s profile information. It can be used to change the profile information for a single group, or to change a particular property for all groups in the library.

```
Usage: ibatch group edit [-s] <group_name>...
Where options are:
-s Use the data on STDIN to edit the meta data.
```

Examples:

To edit the profile information for group ^sales:

```
ibatch group edit "^sales"
```

```
C:\>ibatch group edit "^sales"
----- edit ^sales -----
Enter the GROUP_NAME [^sales]: ^My Sales
C:\>
```

Usage Tips:

- By default, the **edit** command will interactively prompt for input meta data. Where the data to be entered is selectable from a pre-defined list, the list choices are numbered allowing the choice to be conveniently selected.
- When being prompted for meta data, a default value is shown in square brackets []. To accept the default hit enter or return.

group add

The **group add** command creates a new InfoTrove group. It can be run interactively to create a single group, or can be used with file input to create a number of groups. It is an ideal command for initially adding groups to InfoTrove.

```
Usage: ibatch group add [-sd] <group_name>...
Where options are:
-s Use the data on STDIN to edit the meta data.
-d Use default values for meta data.
```

Examples:

To add a single group called "^marketing":

```
ibatch group add "^marketing"
```

```
C:\>ibatch group add "^marketing"
----- add ^marketing -----
C:\>
```

Usage Tips:

- By default, the **add group** command will interactively prompt for input meta data. Where the data to be entered is selectable from a pre-defined list, the list choices are numbered allowing the choice to be conveniently selected.
- When being prompted for meta data, a default value is shown in square brackets []. To accept the default hit enter or return.
- Within InfoTrove all group names start with the ^ character. If you don't specify a ^ character at the start of your group name, Infotrove will add one. Any references to the group after creation will require a ^ character at the start.

COMMAND REFERENCE

Introduction

This section is a reference guide for the ibatch command line interface program. It describes the detailed syntax of the ibatch commands, and discusses advanced topics such as working with files, meta data usage, and the use of standard input and output.

Running ibatch

The ibatch command line interface can be run from any Windows command window, or from any unix or linux shell. The ibatch command can be run by typing ibatch followed by the desired command, operation, and InfoTrove objects.

A summary of the ibatch command syntax is show below.

Note that for information on installing and setting up the ibatch environment refer to [Installation Instructions](#).

IBATCH COMMAND SYNTAX

The ibatch command syntax is:

```
ibatch [-options] <context> <operation> [files...] [folder...]
```

where **<context>** refers to one of the three main ibatch commands - docs, users, or contact.

```
C:\>ibatch -h

Usage: ibatch [-hvRfsd] [-c ibatch.cfg] <context> <operation> [objects..]
Where options are:
  -h Command help.
  -v Version Information.
  -c Load this config file instead of the default $HOME/ibatch.cfg.
  -R Recursive (see individual operation for details)
  -f Force overwrite of files without prompting.
  -s Use stdin, instead of prompting for meta data.
  -d Use default values for meta data.

Where context commands are:
  login   Login in to the InfoTrove server
  docs    list, view, edit, add_doc, add_folder, copy,
          checkout and checkin InfoTrove documents
  user    list, view, add and edit users within InfoTrove
  contact list, view and edit user contact data within InfoTrove
  group   list, view, add and edit groups within InfoTrove
```

OPTION SYNTAX

The command options, indicated by the **[-hvRfsd]** and **[-c *ibatch.cfg*]** arguments, are optional arguments.

Each option can be entered preceded by it's own minus sign "-" option delimiter. For example:

```
ibatch docs add_doc "C:/My Documents/*" "/Docs/Common" -R -d
```

Or several options can be grouped together with a single minus sign "-" option delimiter:

```
ibatch docs add_doc "C:/My Documents/*" "/Docs/Common" -Rd
```

Options can be placed anywhere on the command line after the **ibatch** command itself. The positioning with respect to **<context> <operation> [files...] [folder...]** is flexible.

In general options do not have arguments. The exception is the **-c** option which requires the configuration file name to be specified.

COMMAND ARGUMENT SYNTAX

Mandatory command arguments are indicated via the diagonal brackets **< >**. Optional command arguments are indicated by the square brackets **[]**.

Some commands accept multiple arguments of the same type. For example, **docs add_doc** accepts multiple file names to allow several documents to be added with a single command. Where a command accepts multiple arguments this is indicated in the help description by three periods ... following the argument. For example:

```
C::\>ibatch docs add_doc -h
```

... indicates multiple arguments are supported

```
Usage: ibatch docs add_doc [-Rsd] <local_file|local_folder>... <InfoTrove_folder>
Where options are:
-R Recursive submit of all files and folders below <local_folder>
-s Use the data on STDIN to edit the meta data.
-d Use default values for meta data.
```

CASE SENSITIVITY

All *ibatch* commands, operations, options, file and folder arguments are case sensitive.

QUOTING COMMAND ARGUMENTS

File and folder command arguments should be quoted with "double quotes". For example:

ibatch docs add_doc "C:/My Documents/*" "/Docs/Common" -R

The quotes are required by some shells to protect special non-alphanumeric characters, including the wildcard * operator.

ALLOWED CHARACTERS

A small number of characters are not allowed in certain shells or operating systems, or are not handled transparently via the browser or http transport protocol. As a result, these characters can not be used in certain ibatch commands or on certain InfoTrove forms.

The following lists these restricted characters:

Username	(150 character limit):	V[]: <>"+=;,?*^ space	not allowed
Filenames	(150 character limit):	% ? <# ^~\&+`*/ space	not allowed
Folders	(150 character limit):	% ? <# ^~\&+`*/ space	not allowed
Passwords	(150 character limit):		no restrictions
Doc Folder Type	(150 character limit):	% ? <# ^~\&+`*/, space	not allowed
Doc Number Scheme	(150 character limit):	% ? <# ^~\&+`*/, space	not allowed
Meta Data/Columns	(150 character limit):	alphanumeric and _space	are allowed

Note that any restricted characters used in the filename will be converted to underscores "_" when submitted to InfoTrove.

Object Indexing

Document objects within InfoTrove, such as files and folders, are indexed primarily via the folder/file name path (e.g. "/Docs/Common/Test.doc").

Note that the file name is generally different to the document title displayed in the InfoTrove library. To see the document filename, click on the document title to see the status page information. The filename is listed here. Alternatively, turn on the Advanced column option and scroll across to see the document filename.

User objects are indexed primarily via the username (e.g. ibatch user view "jbloggs") and group objects are indexed primarily via the groupname (e.g. ibatch group view "^sales").

INDEXES ARE CASE SENSITIVE

Object Indexes, such as paths and usernames are case sensitive.

OBJECTS PATH DELIMITER

When indexing documents, the path delimiter can be either the forward slash "/" or the back slash "\" - independent of which operating system or which shell is being used.

ibatch also supports the single dot“.” syntax to represent the current folder, and the double dot “..”syntax to indicate the next highest folder.

For example, to copy a single file called *test.doc* from the InfoTrove /Docs/Common area to the current local disk folder C:/My Documents, the following two commands are equivalent:

```
ibatch docs copy "/Docs/Common/test.doc" "C:/My Documents" and  
ibatch docs copy "/Docs/Common/test.doc" "."
```

To copy the same file to the next higher local folder C:/ the following two commands are equivalent:

```
ibatch docs copy "/Docs/Common/test.doc" "C:/" and  
ibatch docs copy "/Docs/Common/test.doc" ".."
```

Usage Tip:

Some operating system shells use the back slash “\” character to “escape” or “literalize” special characters.

In this case, it is preferable to use the forward slash “/” as the path delimiter to prevent unwanted side effects.

For example, the following syntax would produce undesirable results under Windows 2000 as the second back slash would literalize the last double quote, preventing it being treated as a quote delimiter:

```
ibatch docs list "\Docs\" _____ back slash prevents quote being interpreted  
Error: Can't find any matching objects for \Docs"
```

WILDCARDS

The standard wildcard operator “*” is supported when specifying object indexes. For example, to copy all the files from the InfoTrove /Docs/Common area to your local disk:

```
ibatch docs copy "/Docs/Common/*" "C:/My Documents"
```

Document Number in Filename

If the Doc Number Scheme for an InfoTrove folder is set up to use the document number and revision as part of the stored filename, then this information must be included when indexing a document.

For example, if “Use Number in Filename” is set to “No”, the file *test.doc* may be indexed as:

```
ibatch docs copy "/Docs/Common/test.doc" "C:/"
```

COMMAND REFERENCE

Standard In, Standard Out, Standard Error

In contrast, if “Use Number in Filename” is set to the default “Yes”, the file index would be of the full form:

```
ibatch docs copy "/Docs/Common/Doc007_Rev1_3_test.doc" "C:/"
```

Note that the “Use Number in Filename” option is an Advanced folder option available to Admin users via the **Setup->Doc Number Scheme** menu. For more information on setting up Doc Number Schemes please refer to the InfoTrove user manual.

Standard In, Standard Out, Standard Error

The ibatch program supports shell re-direction for standard input, standard output, and standard error.

The shell re-direction syntax will vary depending on your operating system and shell. However, the following syntax is generally accepted:

```
< or |   for standard input re-direction,  
> or |   for standard output redirection,  
2>       for standard error re-direction.
```

STANDARD INPUT

To use standard input the **-s** option is required to override the ibatch default to interactively prompt for input data.

For example, to change the email address of user jbloggs in-line via the unix csh:

```
echo "EMAIL = 'jbloggs@mycompany.com' " | ibatch user edit "jbloggs" -s
```

Or via a Windows 2000 cmd shell:

```
echo EMAIL = 'jbloggs@mycompany.com' | ibatch user edit "jbloggs" -s
```

Or via an input file:

```
ibatch user edit "jbloggs" -s < jbloggs_data.txt
```

where the file *jbloggs_data.txt* contains the email address string to be changed:

```
EMAIL='jbloggs@mycompany.com'
```

For more information on the syntax to use when editing Meta Data via standard in, refer to [Meta Data](#).

STANDARD OUTPUT

ibatch command output is sent to “standard out” which in most shells defaults to the screen or terminal.

The standard out can be re-directed from the screen to a file using the `>` operator. For example, to list all the library documents to a file called `list_results.txt`:

```
ibatch docs list "/Docs" -R > list_results.txt
```

Usage Tip:

For commands that interactively prompt for user input, the prompts will not be seen if standard out is re-directed. Hence, for these commands, it only makes sense to re-direct standard output, if the standard input option is also being used.

STANDARD ERROR

ibatch error output is sent to “standard error” which in most shells defaults to the screen or terminal.

The standard error output can be re-directed from the screen to a file using the `2>` operator. For example, to list all the library documents to a file called `list_results.txt`, and to send any error messages to a file called `list_errors.txt`:

```
ibatch docs list "/Docs" -R > list_results.txt 2> list_errors.txt
```

Meta Data

ENTERING META DATA INTERACTIVELY

Many of the ibatch commands will prompt the user for the equivalent InfoTrove form or meta data. For example, the following shows a typical transcript when editing user details.

```
C:\>ibatch user edit "jbloggs"
----- edit jbloggs -----
Enter the FULL_NAME [Joe]: Joe J. Bloggs
Enter the Encrypted PASSWD [BEyJQX4gJdb1E]:
  1 - Mail Me _____ List of Choices
  2 - No Mail
Enter the NEW_DOCUMENT number (No Mail) [2]: 1 _____ Enter number for Choice
  1 - Mail If Read Before
  2 - Mail If Edit Access
  3 - Mail If View Access
  4 - No Mail
Enter the DOCUMENT_CHECKIN number (No Mail) [4]: 3
Enter the EMAIL [jbloggs@atrove.com]: _____ Leave blank to accept default
```

Selecting from Meta Data Choices

Where the data to be entered is selectable from a pre-defined list, the list choices are numbered allowing the user to conveniently select the choice.

Default Meta Data Value

The default value is shown in square brackets []. Hitting enter or return without entering any data will accept the current default value.

To accept all default values without being prompted, use the **-d** option.

Entering Multi-Line Data Interactively

When entering data fields that span several lines, such as a new document Abstract, or a checkin Change Log, a single period "." on a new line is used to denote the end of the entry. Note that this is different to the syntax when entering data via an input file.

```

...
Enter the ABSTRACT [Enter a single . on a new line to finish]
[Or enter <CR> to keep the original text]
This is the abstract.
It may take several lines.
To finish the abstract enter a single period . on a new line.
. _____ single period to end multi-line entry
C:\>
```

ENTERING META DATA VIA FILE INPUT

Meta data can be input from a file using the **-s** standard input option. This method is particularly useful when creating automatic scripts, or when adding a large number of documents.

For example, to input the user details for *jbloggs* from a local file called *jbloggs_data.txt*:

```
ibatch user edit "jbloggs" -s < jbloggs_data.txt
```

where the file contents of *jbloggs_data.txt* might look something like:

```

FULL_NAME      ='Joe J. Bloggs'
PASSWORD      ='BEyJQX4gJdb1E'
NEW_DOCUMENT   ='Mail Me'
DOCUMENT_CHECKIN='Mail If View Access'
EMAIL         ='jbloggs@atrove.com'
```

File Meta Data Syntax

When entering Meta Data via a file, the syntax is:

DATA_KEYWORD = 'Value'

The following syntax rules apply:

- (i) Each Data Keyword should be on a separate line in the file.
- (ii) The Value should be enclosed using the single right quotation mark ' .
Do NOT confuse this with the double quote " or the single left quotation mark ` .
- (iii) White space (space, tab etc.) is allowed around the Data Keyword, the equal sign, and the Value.
- (iv) Any text in the file, apart from Data Keywords and the assigned Values are ignored.
- (v) Keywords that you don't have permission to change are ignored.
- (vi) Keywords are case sensitive (i.e. all CAPITALS)

File Meta Data Keywords

The best way to establish which keywords you have permission to change, is to run the command in interactive mode. Then note the Keywords and enter them in the input file.

It is only necessary to enter Keywords in the file corresponding to the values you want to change. The other Keywords can be safely omitted. When running `ibatch`, a warning message will be generated for each of the omitted keywords.

Usage Tip:

A shortcut method for creating the input file is to first run the `view` command and re-direct the output to a file. Then edit the file to change the meta data values, and to remove any unwanted text or Keywords.

For example, to shortcut the file creation used to edit the user details for 'jbloggs' in the previous example:

```
ibatch user view "jbloggs" > jbloggs_data.txt
```

COMMAND REFERENCE

Meta Data

The resultant output file, *jbloggs_data.txt*, will look something like:

```
----- jbloggs -----
ID                        ='3'
FULL_NAME                 ='Joe J. Bloggs'
USER_NAME                 ='jbloggs'
PERMISSION                ='Normal'
PASSWORD                  ='BEyJQX4gJdb1E'
NEW_DOCUMENT              ='Mail Me'
DOCUMENT_CHECKIN          ='Mail If View Access'
EMAIL                     ='jbloggs@atrove.com'
GROUPS                    ='^all'
```

The output file contains several Data Keywords that are not editable by 'jbloggs' as a 'Normal' user. For example, jbloggs can not edit the **PERMISSION** level although he can view it.

The output file also contains a system **ID** Keyword, and a text header **-----jbloggs-----**, which are not valid inputs for the **user edit** command.

The invalid Keywords and text can be removed from the file before input to the edit command. Or they can be left in, as they will be simply ignored by the *ibatch* program.

Applying Meta Data to Multiple Command Arguments

When executing a command with multiple arguments, such as when using the wildcard to denote multiple files, any meta data on standard in will apply to all arguments.

For example, to change the mail option of all users, the following command could be used:

```
ibatch user edit * -s < mail_options.txt
```

Where the *mail_options.txt* file contains the following Data Keyword entries:

```
NEW_DOCUMENT              ='Mail Me'
DOCUMENT_CHECKIN          ='Mail If View Access'
```

Entering Multi-Line File Data

When entering data fields that span several lines, such as a new document Abstract, or a checkin Change Log, a single right quote ' on a new line is used to denote the end of the entry. Note that this is different to the syntax when entering interactively via the command line.

```
...
ABSTRACT          ='This is the abstract
it may take several lines.
To finish the abstract enter a single right quote on a new line to finish.
'
```

single right quote to end multi-line entry

Return codes

Each **ibatch** command outputs a shell return code which can be tested to indicate if the command was successful.

The shell return codes are:

0=success

1=warning

2=non-fatal error (typically returned when one part of a multi-function command fails which does not prevent the next part being executed)

3=fatal error

When multiple command arguments are processed, the error code returned is the highest number produced by processing the individual arguments.

The way the shell return code is accessed will depend on your operating system and the particular shell being used. For example, in the Windows 2000 cmd shell, the return code is recorded in the variable "errorlevel" which can be tested using the "if" command. Whereas, in the unix csh, the return code is recorded in the shell "status" variable.

Please consult your shell documentation for details on how to access and test the shell return code.

Permissions and Security

When running **ibatch**, permission and security features function in the same way as InfoTrove. Every time an **ibatch** command is run, the encrypted username and password are retrieved from the configuration file and passed to the InfoTrove server. If your user credentials entitle you to perform a certain function within InfoTrove, you will be able to perform that same function using the equivalent **ibatch** command.

Mail Considerations for Bulk Operations

When performing bulk operations, such as adding an entire folder tree of documents to InfoTrove, a large number of new document email messages will be generated.

It may be desirable to temporarily turn off these email messages when performing bulk operations.

For example, to turn off the mail option for all users, the following command could be used:

```
ibatch user edit * -s < mail_options.txt
```

Where the *mail_options.txt* field contains the following Data Keyword entries:

```
NEW_DOCUMENT          ='No Mail'  
DOCUMENT_CHECKIN     ='No Mail'
```

Object Locks

When running *ibatch*, object and file locks function in the same way as InfoTrove. For example, if a file is checked out, or some meta data is being edited, then other users will be prevented from editing the corresponding object until the operation is completed.

Usage Tip:

When using *ibatch* scripts to automatically test the value of object data (e.g. meta data) before performing an action command (e.g. edit operation), care should be taken to ensure the data does not change between the test and action commands.