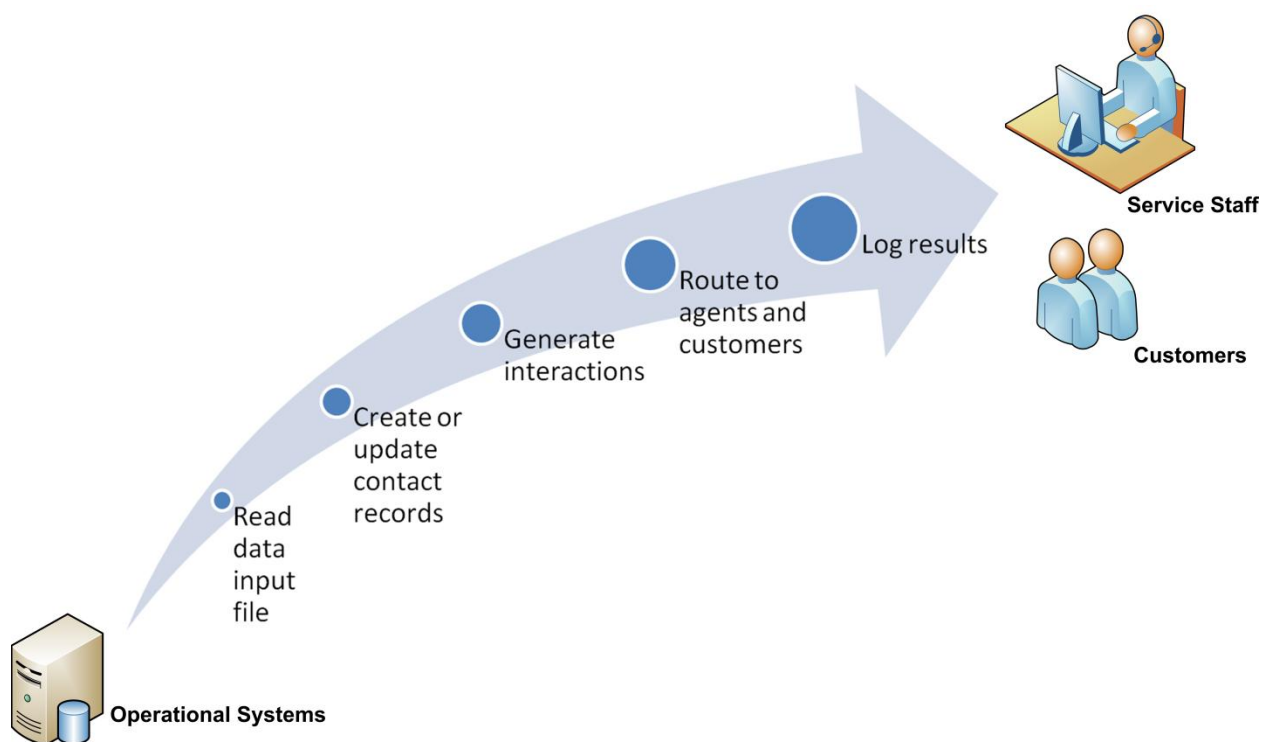


The iService®

Batch Utility

Utilities for integrating iService with ERP and other operational systems



Copyright © 2009-2011 One-to-One Service.com, Inc. All Rights Reserved

Table of Contents

| | |
|--|-----------|
| BATCH UTILITY OVERVIEW | 4 |
| BATCH FORM SUBMISSION UTILITY | 5 |
| THERE ARE THREE STEPS REQUIRED WHEN USING THE BATCH FORM SUBMISSION UTILITY. | 5 |
| COMMAND LINE UTILITY WITH FORMS | 8 |
| COMMAND LINE UTILITY WITHOUT FORMS | 9 |
| BATCHUTILITY.EXE | 9 |
| INPUT_FILENAME.CSV | 9 |
| WEBSERVICE URL | 9 |
| AGENT_USER_NAME | 9 |
| PASSWORD | 9 |
| RENAMEINPUT | 9 |
| - AGENTEMAIL -TICKET -NOTEPUBLIC -NOTEPRIVATE | 10 |
| INPUT FILE REQUIREMENTS | 11 |
| CONTACT PROPERTIES | 11 |
| INTERACTION PROPERTIES | 11 |
| REQUIRED AGENT EMAIL PROPERTIES | 12 |
| REQUIRED NOTE PROPERTIES (PUBLIC AND PRIVATE NOTES) | 12 |
| REQUIRED TICKET PROPERTIES | 13 |
| ERROR LOGGING | 14 |
| FILE PROCESSING ERRORS | 14 |
| RECORD PROCESSING ERRORS | 14 |
| CONTACT IMPORT ERRORS | 14 |
| AGENT EMAIL ERRORS | 14 |
| AUTOMATION WITH SCHEDULED TASKS | 15 |
| ERROR CODES AND TROUBLESHOOTING | 16 |



| | |
|--|-----------|
| FILE PROCESSING ERRORS | 16 |
| HTTPS VS HTTP | 16 |
| NO WEBSITE FOUND | 16 |
| INCORRECT FILENAME OR PATCH | 16 |
| INPUT FILE IS OPEN OR LOCKED | 16 |
| INCORRECT LOGIN OR PASSWORD | 16 |
| MISSING DLL | 16 |
| INVALID COLUMN HEADERS | 17 |
| RECORD PROCESSING ERRORS | 17 |
| INCORRECT VALUE FOR BOOLEAN PROPERTY TYPE | 17 |
| INVALID TOPIC ID | 17 |
| INVALID SEGMENT ID | 17 |
| MISSING REQUIRED FIELDS | 18 |
| EMAIL ADDRESS DOES NOT BELONG TO CUSTOMER | 18 |
| ATTEMPTING TO UPDATE AGENT RECORDS / LOGIN NAMES ARE READ-ONLY | 18 |
| PROCESSING DETAILS | 19 |

Batch Utility Overview

iService includes two types of utilities that can be used for importing contacts and generating interactions from an input file. The *iService Batch Form Submission* utility can be used to read a file of information and process it through a predefined iService form. This utility includes a graphical user interface and is the recommended method for importing contacts or generating agent e-mail messages from a CSV file.

In addition to the *iService Batch Form Submission* utility, iService includes a command line utility that performs similar actions but without a predefined form. Because the command line utility does not use a predefined iService form, it requires more definition within the input file itself.

The batch form submission utility and command line utility can both be used to perform the following actions based on an input file.

1. Create new contacts
2. Set, update, or remove properties for contacts (e.g., name, email, address, phone, etc.)
3. Send agent emails to contacts and set interaction properties within the email
4. Create notes within contact history
5. Create tickets and queue them for agent handling

The utilities read a comma separated value(CSV) text file that contains the contact information to import and the details related to the interactions to create (agent email, notes, or tickets). The Command Line Utility is designed to be run at the command line or included in a batch file to automate routine execution on a predetermined schedule. The Batch Form Submission utility includes a graphical user interface and is significantly faster than the command line utility.

The utilities perform a variety of error checks to validate the input file. However, it is recommended that you ensure users are familiar with its operation and limitations before making it broadly available. Also, you should test the utility against a test file with limited data before using it on a large scale project. The source code for this utility is provided so that you can modify or extend its functionality. You can also use it as the baseline for other projects.

This user guide describes how to use both utilities.

Batch Form Submission Utility

There are three steps required when using the batch form submission utility.

1. Create the form to which you will submit your information.

The utility reads a row from the input file and submits it to the specified form. Therefore, the first step is to create a form that will accept this input. The layout of the form is not relevant since it will only be used by the utility. An example of a simple form for importing contact information is shown below. This form accepts input for email, first name, last name, and company.

Form Body

```
<html>
<body>
<form method="POST">

$input -email -id'email'$<br />
$input -contactProperty1'firstname' -id'firstname'$<br />
$input -contactProperty3'lastname' -id'lastname'$<br />
$input -contactProperty7'company' -id'company'$<br />

<input type="submit" id="ok" name="ok" value="Submit" /><br />

</form>
</body>
</html>
```

Form Submit Action

Find/Create Contact

Find By Email: OR Find By Contact ID:

Property Group: *Specified with \$input ... -group'xxx'\$

Leave existing values when properties in form are blank.

Result Name: *Can be referenced in other steps as \$result -id'name'\$

For more information on creating forms, view the [iService Forms User Guide](#).

2. Add column labels to your input file that match the form.

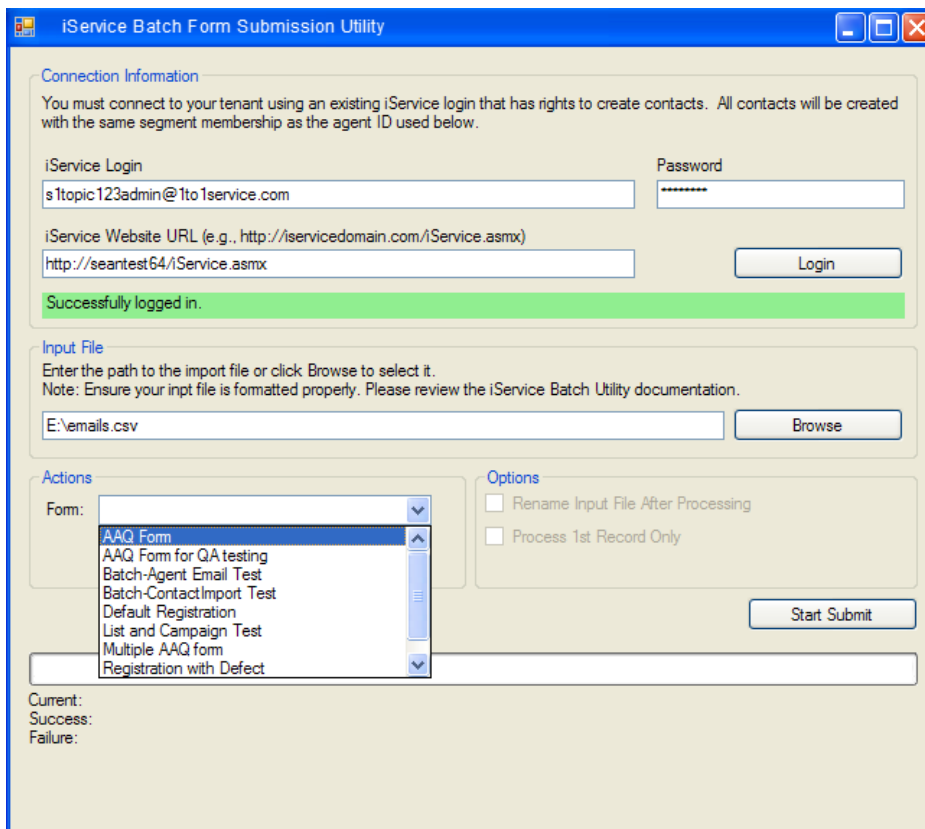
iService will match the column headings of your CSV file to the input fields on the form. The order of the columns does not matter. The column headings for the form created in step 1 above would be as follows. Just append "form-" to the id of the input field from the form.

| | A | B | C | D |
|---|-----------------|----------------|---------------|------------------|
| 1 | Form-email | Form-firstname | Form-lastname | Form-company |
| 2 | user@domain.com | John | Smith | ACME Corporation |

3. Run the iService Batch Form Submission utility as described below.

The Batch Form Submission utility reads a CSV file and submits each role of information to a predefined iService form. The iService form defines the actions taken, such as creating contacts or sending agent e-mail messages.

The first step when using the utility is to complete the login section and enter the credentials of the agent that is submitting the forms. After clicking the login button, you should see the successfully logged in message. The second step is to select the input file either by browsing your computer for it or typing the path directly into the input box. The third step is to select the form to which you are submitting your input file.

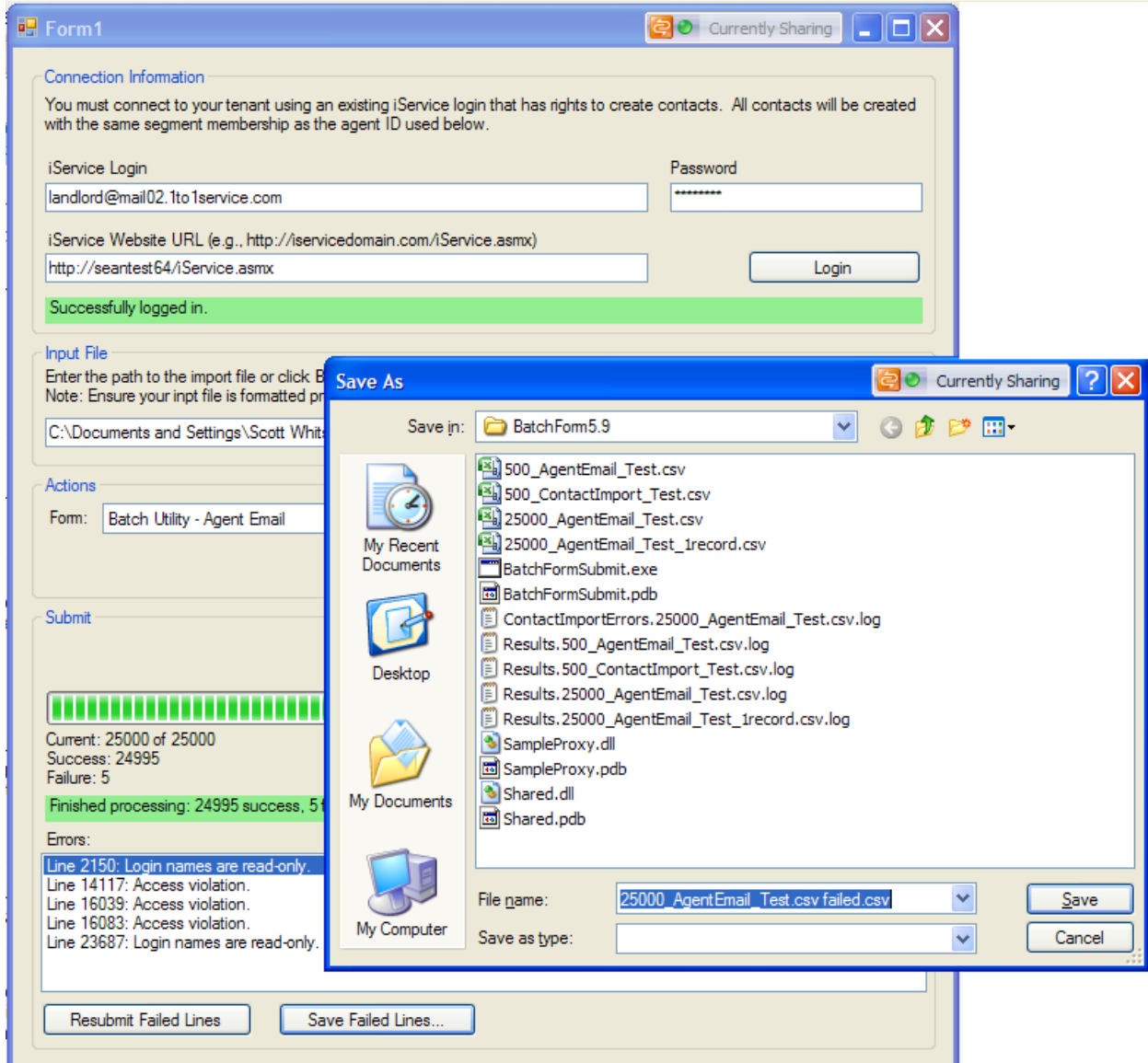


Step 1 - Login

Step 2 - Select the input file

Step 3 - Select the form that will accept the input.

If errors are encountered during processing, they will be displayed in a text box below the progress bar. Two buttons will display with options to attempt reprocessing, or saving the failed records to a file for later processing. We recommend that you first attempt to reprocess the records, and then save the failed records as shown below.



Command Line Utility with Forms

There are two ways to use the command line utility: with forms and without forms. Using forms provides a significant increase in processing speed, and simplifies the requirements of the CSV input file.

To use the command line utility with forms, follow the first two steps outlined in the Batch Form Submission Utility section above. Once the form and input file are prepared, use the following command to execute the utility.

Usage (note: remove " from command line):

```
BatchUtility "input_filename.csv" "webservice URL" "agent_user_name"  
"password" -formid "id#"
```

```
[-renameinput]
```

The `-formid` parameter tells the utility to process the CSV input file through iService forms. The id of the iService form can be found from the Admin Tools – Forms page.

Command Line Utility without Forms

The utility can be invoked at the command line, and can thus be used in an automated routine. The command line parameters are as follows:

```
Usage: BatchUtility "input_filename.csv" "webservice URL"  
"agent_user_name" "password"
```

```
[-renameinput]
```

```
[-agentemail | -ticket | -notepublic | -notepivate]
```

BatchUtility.exe – This is the name of the utility executable.

Input_Filename.csv – This is the full name and path to the input file that will be processed. For instance, if the file is located in the C:\import directory the parameter might be
c:\import\pastdueaccounts.csv.

Webservice URL – The webservises URL path will define the tenant against which the utility will run. For on-premises clients, this might be a test tenant or a production tenant. The URL path must include the full name of the .asmx file. For example, your webservises URL might be
https://1to1service.iServiceCRM.com/iservice.asmx.

Agent_User_Name – This is the UserID of the agent that is used to create the contacts and generate interactions. It will be recorded in history for each interaction created. Also, the segment access of this agent will be used when creating new contacts, and will be validated when agent emails are sent. Therefore, you must ensure the agent access for this User ID is appropriate for your mailing.

NOTE: Contact replies to your agent email are automatically reserved for the agent that sent the email. If you do not want replies to your mailing to be reserved for the UserID used during execution, you must set that agent's properties for Empty Inbox to TRUE. This will cause any replies to the mailing to be re-queued and available for other agents.

Password – This is the iService password for the User ID above.

The command line usage allows you to create a batch file that contains multiple instances of the batch mailings, each with their own parameters. This batch file can then be executed on a predefined schedule using a Windows Task Scheduler.

RenameInput – This is an optional parameter that can be added to the command to rename the input file upon completion. The file will be renamed as shown below. Without this parameter the original file will be left in place.

```
originaldir\YYYYMMDD originalname.ext
```

- **agentemail** | -**ticket** | -**notepublic** | -**noteprivate** – These parameters indicate whether the input file contains agent emails, tickets, public notes, or private notes.

Input File Requirements

When used without forms, the command line utility is designed to read a comma separated value file that contains contact information and interaction details. The contact properties imported are based on the name of each property, while interaction properties are based on the propertyID from the iService database.

The first line of your input file MUST specify the details of each column in the file (i.e., column header). And, the first column that includes a contact property MUST specify the identifier for the contact record. The contact identifier is usually the Login (Email Address) of the contact, but if you use another contact property as a unique identifier (e.g., account number) that could also be used. However, beware that Login and Contact ID are the only database enforced unique field for a contact.

The format for the Login/Email address must be one of the following:

`contact-login` **or** `contact-Login` (email).

The format for column headings is as follows:

Contact Properties

`contact-(description) DISPLAY NAME`

Contact indicates the property is a contact property. Description is only used for multi-value properties that have a description (e.g., phone number might include home or mobile). The Display Name is exactly as shown within the iService Customer Info>Contacts>Details tab. Some examples of contact property headings are shown below.

`contact-(home) Phone`

`contact-Company`

Interaction Properties

`interaction-(description) ID`

Interaction indicates the property is an interaction property. Description is only used for multi-value properties that have a description. The ID is the actual property ID from the database, and can be found in the Admin Tools>Segment>Interaction Properties tab. Some examples of interaction property headings are shown below.

`interaction-(department)2 ~` this indicates an interaction property whose ID is 2. However, this property allows more than one value and each value has a label. An example of this might be tracking billable minutes by department. The property captures the number of billable minutes, but a label is used to indicate the corresponding department.

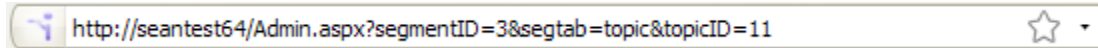
`interaction-4 ~` this indicates an interaction property whose ID is 4.

Input File Requirements When Used without Forms

Required Agent Email Properties

Agent emails require the following values, and the column headings MUST match the definition below or file processing will fail.

- `emailtopicID` – The ID number of the topic for the message. This can be found in the Admin Tools>Segments>Topics tab by highlighting the desired topic. The topic ID is displayed in the address bar of the web browser when a topic is selected. In the example below, topic ID 11 is selected.



- `emailbody` – The body of the email. Typically, this will be a stock response that is predefined rather than a full text body. If a stock response is used, it must be entered in the standard variable format as `$stock#$` or `'$stock#' name' $`. The # represents the ID of the stock response, and the name represents the display name of the stock response. In all cases, the ID of the stock response must be present and if a name is used it must be inside single quotes.
- `emailsegmentID` – The ID of the segment in which the agent email is sent.
- `emailmailboxID` – The ID of the mailbox used to send the email. It must be a mailbox within the selected segment.
- `Emailsubject` – The subject line of the agent email message.
- `Emailaddress` – The email address of the recipient. This email address must exist as a Login (Email) property value, but will be created by iService if it does not already exist.

Optional Fields

- `AddRef` – If this field is included and the value is FALSE, the message Ref# will not be included in the subject line of the agent email when sent. If this value is not included, the standard message format including the Ref# in the subject line will be used.
- `Emailnote` – Information stored in the Private Comments section of the agent email. This is an optional parameter.

Required Note Properties (Public and Private Notes)

Adding a note to contact history requires most of the same values as Agent Emails.

- `emailtopicID` – same as agent email.
- `emailbody` – same as agent email.
- `Emailnote` – same as agent email.
- `emailsegmentID` – same as agent email.
- `Emailsubject` – The subject line of the note.

The following fields are **NOT** used when creating notes for history.

- `Emailaddress`

Input File Requirements When Used without Forms

- emailmailboxID

Required Ticket Properties

Creating a ticket requires most of the same values as Agent Emails.

- emailtopicID – same as agent email.
- emailbody – same as agent email.
- Emailnote – same as agent email.
- emailsegmentID – same as agent email.
- Emailsubject – The subject line of the ticket.
- Emailaddress – same as agent email.

The following fields are **NOT** used when creating Tickets.

- emailmailboxID

Error Logging

There are two types of errors that might occur in processing these files: File processing errors and record processing errors. These errors are logged as follows.

File Processing Errors

There are various conditions that will cause the overall utility to fail and not process any records. These include an invalid tenant, invalid file format/path/name, incorrect userID and password etc. Also, the field identifiers in the input file might be incorrect and thus none of the records can be processed. These types of errors are logged into the File Processing Results log file.

The naming convention for this file is `FileProcessing.inputfilename.log`.

Two examples of these errors are as follows:

```
At: 2009-12-22 08:53:05
```

```
Error processing TestFile.csv: Login failure: Login name/password not recognized.
```

```
At: 2009-12-22 09:11:03
```

```
Error processing TestFile.csv: Line 1: Invalid column headers: unknown field specified: interaction-AmountDuee
```

As shown above, the log file is appended with each additional error.

Record Processing Errors

Record processing errors are divided into two categories: Contact Import Errors and Agent Email Errors.

Contact Import Errors – If records within the input file can't be processed because of errors relating to the contact create or update process, details of those errors will be written to a contact import errors log file. This file will include the row# for each error, along with a description of the error. This file will ONLY contain error records.

The naming convention for this file is `ContactImportErrors.inputfilename.log`.

Agent Email Errors – If records within the input file passed the contact validation but did not pass the interaction creation validation, they will be written to the agent email error log. This file will include the row number of the record along with the error reason, and will ONLY contain error records.

The naming convention for this file is `AgentEmailErrors.inputfilename.log`.

Automation with Scheduled Tasks

The Batch Utility was created as a command line program so it can be easily invoked from within a batch file. The Windows Server Task Scheduler can then be used to run the batch file according to whatever schedule is desired.

An example of the contents of such as batch file is shown below.

```
Batchutility.exe TestFile1.csv https://1tol.iservicecrm.com/iservice.asmx agent4utility  
password -renameinput  
  
Batchutility.exe TestFile2.csv https://1tol.iservicecrm.com/iservice.asmx agent4utility  
password -renameinput
```

This batch would process against two files, TestFile1 and TestFile2, located in the same directory as the batch file. The file can be located in any directory that is accessible from the location of the batch file.

Using the task scheduler and a batch file allows you to generate files using a reporting tool, such as SQL Server Reporting Services, and save those files into the directory specified by the batch file. The file will then be executed by the Windows Task Scheduler as required (daily, weekly, etc.) and the files will be automatically processed.

The optional `-renameinput` parameter will rename the file upon completion, thus allowing your report writing software to place a new file into the directory without a file naming conflict.

Error Codes and Troubleshooting

The IBatchutility.exe program uses a variety of iService web services to complete its tasks. If errors are encountered these web services return a variety of error codes. The utility logs these errors in log files described in section Error Logging above. Some of the more common errors are explained below. When a File Processing Error occurs, not of the records will be processed. When a Record Errors occurs, the offending record will be skipped but the other records in the file will be processed.

File Processing Errors

HTTPS vs HTTP

This error message indicates that the iservice.asmx page was not found in the location specified. If the website requires https and your path only includes http, you will receive the following error.

```
Error processing filename: Login failure: The request failed with HTTP status 404: Not Found.
```

No Website Found

If you type the web site address wrong you will receive the following error.

```
Error processing filename: Login failure: The request failed with an empty response.
```

Incorrect Filename or Patch

If the file for processing is not at the location or is misspelled, the following error is generated.

```
Error processing filename: Failed to open file: Could not find file 'path/filename'.
```

Input File is Open or Locked

If the csv file you are using is open in Excel or otherwise locked, you will get the following error.

```
Error processing filename: Failed to open file: The process cannot access the file path/filename' because it is being used by another process.
```

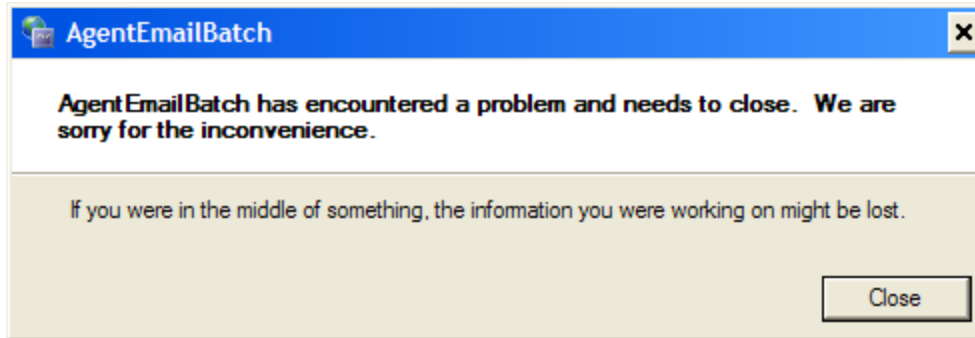
Incorrect Login or Password

If you enter the username or password incorrectly, you will get the following error.

```
Error processing filename: Login failure: Login name/password not recognized.
```

Missing DLL

The AgentEmailBatch.exe utility requires the shared.dll file to be in its directory. If this dll is not present, you will get the following Windows error display:



After closing this window, you will see an error similar to the following in your DOS Command Line.

```
Unhandled Exception: System.IO.FileNotFoundException: Could not load file or assembly 'Shared, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null' or one of its dependencies. The system cannot find the file specified.
```

```
File name: 'Shared, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null' at AgentEmailBatch.Program.Main(String[] args)
```

Invalid Column Headers

The batch utility uses the column headings to identify the values for each record. If you misspell a column heading or use a value that is invalid, you will receive an error like the following.

```
Error processing filename.csv: Line 1: : Invalid column headers: unknown field specified: ContactIDs.
```

Record Processing Errors

Incorrect Value for Boolean Property Type

This error message indicates that the record contained a value that is not true, false, or empty. If the property type you are using for the interaction property is Boolean, then the value for the record must be true, false, or left empty. You will receive an error like the following.

```
Error processing filename.csv: Line 2: 'addref' column value must be 'true', 'false', or empty.
```

Invalid Topic ID

Every interaction must have a topic ID that corresponds to a valid topic within your iService tenant. If you have an invalid topicID value, you will receive an error like the following.

```
Error processing filename.csv: Line 3: Error sending agent email to contact ID 37: Invalid topic ID specified: 100.
```

Invalid Segment ID

Every interaction must have a segment ID that corresponds to a valid segment within your iService tenant. If you have an invalid segment ID value, you will receive an error like the following.

```
Error processing filename.csv: Line 4: Error sending agent email to contact ID 38: Invalid segment ID specified: 6.
```

Missing Required Fields

Similar to the invalid value error above, you will receive an error for any record that is missing one of the required fields. If you missing values for required fields, you will receive an error like the following.

```
Error processing filename.csv: Line 8722: Missing one or more required fields: emailbody, emailmailboxID, emailsegmentID, emailsubject, emailaddress, or emailtopicID.
```

Email Address Does Not Belong to Customer

The process for sending an email or creating a ticket requires selection of one of the contacts email addresses. Since the contact may have more than one email address, it is not necessarily the same as the Contact-Login. If have a mismatch between the contacts login and their email address, you will receive an error like the following.

```
Error processing filename.csv: Line 4: Error sending agent email to contact ID 141: Specified email address does not belong to the customer.
```

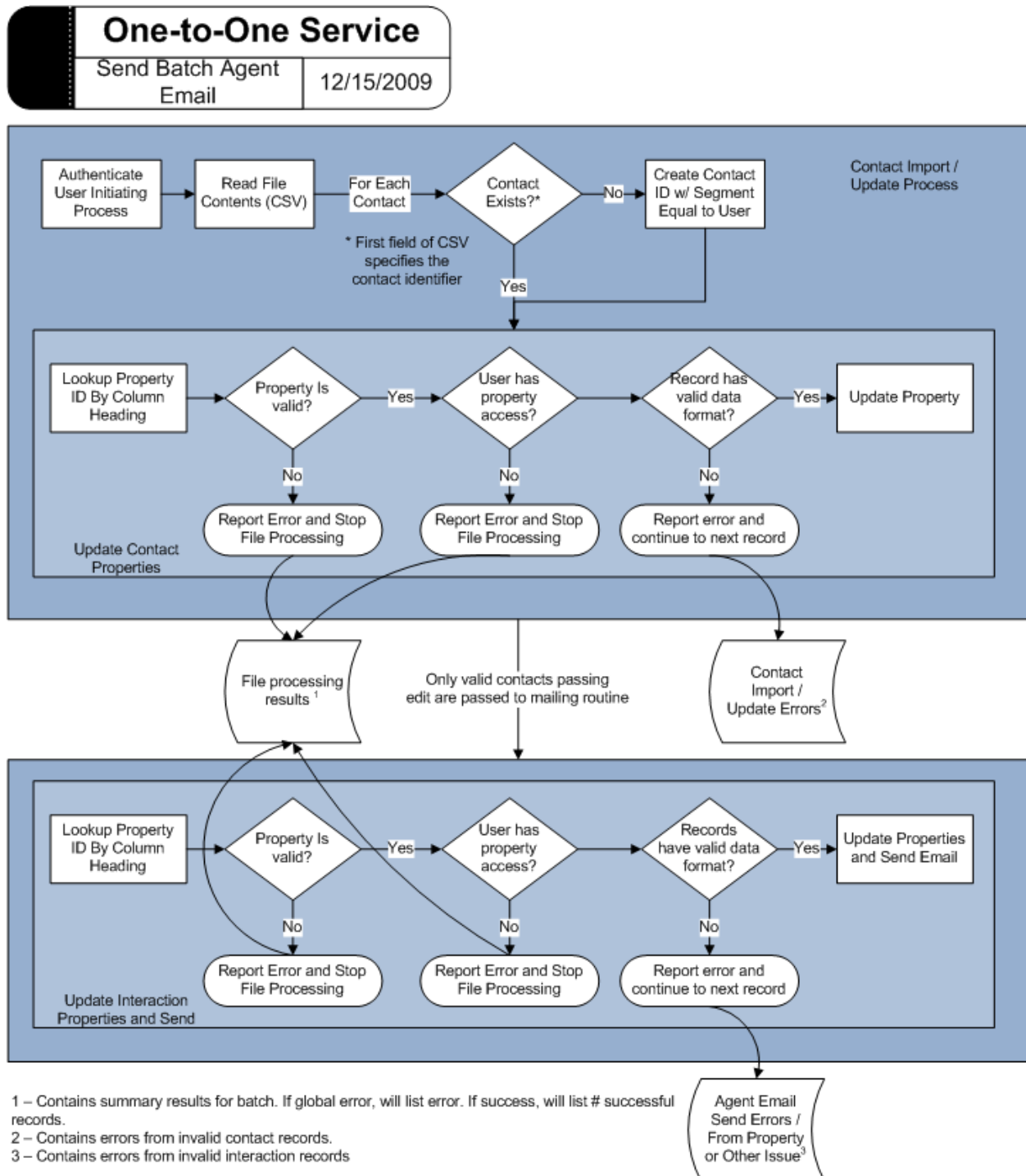
Attempting to Update Agent Records / Login Names are Read-Only

Agents are not allowed to change the login of other agents unless they have sufficient access rights. If your input file contains agents that have rights equal to or above the accounts used with the batch utility, you will receive an error like the following.

```
Error processing filename.csv: Line 2: Error updating properties for contact ID 2: Login names are read-only.
```

Processing Details

As shown below, the utility is comprised of two primary functions: Importing and updating contact information, and generating interactions (e.g., agent email, notes, and tickets). The diagram below provides a basic overview of the logic used within the application for agent emails.



The first routine within the utility ensures the contact record exists and updates the contact properties that are included within the input file. This routine validates all of the contact properties included within the input file and logs error for any records that cannot be updated. The second routine builds the agent email (or other type of interaction) and validates all of the required interaction properties. If the input file does not contain all of the required fields for the specified type of interaction (see the Input File Requirements section below), the second portion of the routine is skipped and the operation is limited to contact import and update.

Since all of the actions performed by the utility are based on a contact, it's important that the utility be run with the desired iService segment access. The lookup function that attempts to determine whether or not a contact exists uses the iService contact search web service. If the login used with the utility does not have access to the contact, the search results will be blank and the utility will attempt to create a new contact. However, the contact creation will fail because the contact login already exists. Also, when new contacts are created, they will be assigned to the same segments for which the running login has segment access. For instance, if you run the utility using a SuperUser login all new contacts will be assigned membership to all business segments.