## Knowledge Base Article #ETMD921, Rev A Importing Directory Listings: Modifying the Format for the Import File

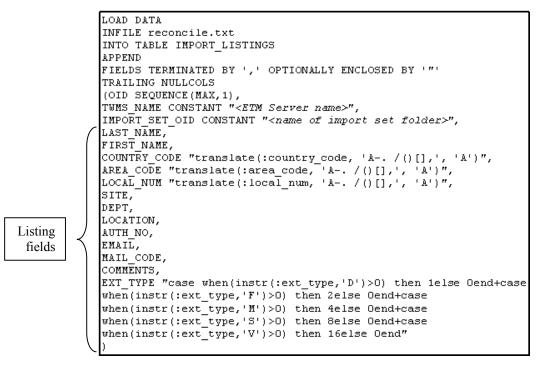
- **Synopsis** When you import Listings from a text file into the Directory Manager, the text file must be formatted in a certain way for successful import. The required format is determined by a control file in the ETM<sup>®</sup> System installation directory that is used by the SQL\*Loader application. This article describes how to edit the file to modify the required format for the imported file(s).
- More Information When you create a new Import Set, a system-named Import Set directory is created in the ps\directory\import\_sets directory of the ETM System installation directory—for example, C:\Program Files\SecureLogix\ETM \ps\directory\import\_sets\0102d-7fffffff-7fffffec 00000002. A text file in this directory named import\_set\_details.txt identifies the Import Set by the name you gave it in the GUI.

In each Import Set directory, the file **reconcile.ctl** defines the required format for imported file(s).

## Create an Import Set To create an Import Set

- 1. In the Directory Manager tree pane, right-click **Import Sets**, then click **New**. The **Create Import Set** dialog box appears.
- Type a name for the Import Set of up to 32 characters. The name can include letters, digits, spaces, and the following special characters:
   & ().!'+=
- 3. Click **OK**. The Import Set appears in the **Import Sets** node of the Directory Manager tree pane.
- 4. Edit the Control File as described in the next section.

*Edit the Control File* When a new Import Set is created, the following is the default format of the **reconcile.ctl** file:



The order in which the Listing fields appear in this file is the order in which they must appear in the text file to be imported.

## To edit the reconcile.ctl file

1. Open reconcile.ctl in a text editor.

**Note:** Making a backup copy of any file you intend to edit is good practice.

2. Arrange the Listing fields in the order that they appear in the text file to be imported.

For example, if the fields in the text file are arranged as follows:

COUNTRY\_CODE, AREA\_CODE, LOCAL\_NUM, LAST\_NAME, FIRST\_NAME, SITE, DEPT, LOCATION, AUTH NO, EMAIL, MAIL CODE, COMMENTS, EXT TYPE

you would arrange the fields as shown in the illustration below.

**IMPORTANT** Do not edit any text in the file except the delimiter (see the note on the following page). Do not rearrange the first 9 lines.

Do not rearrange the first 9 lines LoAD DATA INFILE reconcile.txt INTO TABLE IMPORT_LISTINGS APPEND FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' TRAILING NULLCOLS (OID SEQUENCE (MAX, 1), TWMS_NAME CONSTANT " <etm name="" server="">", IMPORT SET_OID CONSTANT "<reame folder="" import="" of="" set="">", COUNTRY_CODE "translate(:area_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCATION, SITE, DEPT, LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then lelse Oend+case when(instr(:ext_type,'Y')&gt;0) then 2else Oend+case when(instr(:ext_type,'Y')&gt;0) then 3else Oend+case when(instr(:ext_type,'V')&gt;0) then 16else Oend" )</reame></etm>		
Do not rearrange the first 9 lines INTO TABLE IMPORT_LISTINGS APPEND FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' TRAILING NULLCOLS (OID SEQUENCE (MAX, 1), TUMS NAME CONSTANT " <etm name="" server="">", IMPORT_SET_OID CONSTANT "<name folder="" import="" of="" set="">", COUNTRY CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCAT NUM "KINAME, FIRST_NAME, SITE, DEPT, LOCATION, ANTL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then lelse Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</name></etm>	Ć	LOAD DATA
Do not rearrange the first 9 lines In the line that begins FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"' TRAILING NULLCOLS (OID SEQUENCE (MAX, 1), TUMS NAME CONSTANT " <etm name="" server="">", IMPORT_SET_OID CONSTANT "<name folder="" import="" of="" set="">", COUNTRY_CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL NUM "translate(:local_num, 'A /()[],', 'A')", LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</name></etm>		INFILE reconcile.txt
<pre>rearrange the first 9 lines</pre> FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ''' TRAILING NULLCOLS (OID SEQUENCE (MAX, 1), TWMS NAME CONSTANT " <tame>", IMPORT SET OID CONSTANT "<name folder="" import="" of="" set="">", COUNTRY CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LAST NAME, FIRST NAME, FIRST NAME, FIRST NAME, SITE, Values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with 't.</name></tame>		INTO TABLE IMPORT LISTINGS
TRAILING NULLCOLSfirst 9 linesIn the line that beginsFTELDS TERMINATED BY, the comma (, ) indicates that the text file you will import is TAB delimited, replace the comma with \t.TRAILING NULLCOLS (OID SEQUENCE (MAX, 1), TWMS_NAME CONSTANT " <etm name="" server="">", IMPORT_SET_OID CONSTANT "<name folder="" import="" of="" set="">", COUNTRY_CODE "translate(:country_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then ielse 0end+case when(instr(:ext_type,'S')&gt;0) then 8else 0end+case when(instr(:ext_type,'S')&gt;0) then 8else 0end+case</name></etm>	Do not	APPEND
<pre>first 9 lines first 9 lin</pre>	rearrange the $\prec$	FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
<pre>(01D SEQUENCE (MAX,1), TWMS_NAME CONSTANT "<ftm name="" server="">", IMPORT_SET_OID CONSTANT "<name folder="" import="" of="" set="">", COUNTRY_CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LAST_NAME, FIRST_NAME, SITE, DEPT, LOCALTION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then lelse 0end+case when(instr(:ext_type,'S')&gt;0) then 8else 0end+case</name></ftm></pre>	•	TRAILING NULLCOLS
<pre>In the line that begins FIELDS TERMINATED BY, the comma (,) indicates that the text file you will import is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t. </pre>	first 9 lines	(OID SEQUENCE (MAX, 1),
<pre>COUNTRY_CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LAST_NAME, FIRST_NAME, FIRST_NAME, SITE, SITE, Values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t. COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>		TWMS NAME CONSTANT " <etm name="" server="">",</etm>
<pre>COUNTRY_CODE "translate(:country_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LAST_NAME, FIRST_NAME, FIRST_NAME, SITE, SITE, Values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t. COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>		IMPORT SET OID CONSTANT " <name folder="" import="" of="" set="">",</name>
<pre>AREA_CODE "translate(:area_code, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LOCAL_NUM "translate(:local_num, 'A /()[],', 'A')", LAST_NAME, FIRST_NAME, FIRST_NAME, SITE, DEPT, values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t.</pre>		
<pre>In the line that begins FIELDS TERMINATED BY, the comma (,) indicates that the text file you will import is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t. LOCAI_NUM "translate(:local_num, 'A /()[],', 'A')", LAST_NAME, FIRST_NAME, SITE, DEPT, LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'S')&gt;0) then 3else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>		
<pre>FIELDS TERMINATED BY, the comma (,) indicates that the text file you will import is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t.</pre> LAST NAME, SITE, DEPT, LOCATION, AUTH_NO, EMAIL, NAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else Oend+case when(instr(:ext_type,'S')>0) then 2else Oend+case when(instr(:ext_type,'S')>0) then 8else Oend+case	In the line that begins	
<pre>comma (, ) indicates that the text file you will import is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t.</pre> FIRST_NAME, SITE, DEPT, LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else Oend+case when(instr(:ext_type,'S')>0) then 2else Oend+case when(instr(:ext_type,'S')>0) then 8else Oend+case	FIELDS TERMINATED BY, the	
<pre>the text file you will import is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t.</pre> SITE, UCCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else 0end+case when(instr(:ext_type,'F')>0) then 2else 0end+case when(instr(:ext_type,'S')>0) then 4else 0end+case	comma (,) indicates that	
<pre>DEPT, is a comma-separated values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t.</pre> DEPT, LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else 0end+case when(instr(:ext_type,'S')>0) then 2else 0end+case when(instr(:ext_type,'S')>0) then 4else 0end+case when(instr(:ext_type,'S')>0) then 8else 0end+case		
LOCATION, values (CSV) file. If the text file you want to import is TAB delimited, replace the comma with \t. LOCATION, AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else Oend+case when(instr(:ext_type,'F')>0) then 2else Oend+case when(instr(:ext_type,'S')>0) then 4else Oend+case when(instr(:ext_type,'S')>0) then 8else Oend+case		
AUTH_NO, text file you want to import is TAB delimited, replace the comma with \t. AUTH_NO, EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')>0) then 1else Oend+case when(instr(:ext_type,'F')>0) then 2else Oend+case when(instr(:ext_type,'S')>0) then 4else Oend+case when(instr(:ext_type,'S')>0) then 8else Oend+case	•	
<pre>EMAIL, is TAB delimited, replace the comma with \t. EMAIL, MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'F')&gt;0) then 2else Oend+case when(instr(:ext_type,'K')&gt;0) then 4else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>	( <i>'</i>	
<pre>MAIL_CODE, the comma with \t. MAIL_CODE, COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'F')&gt;0) then 2else Oend+case when(instr(:ext_type,'K')&gt;0) then 4else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>		
<pre>the comma with \t. COMMENTS, EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'F')&gt;0) then 2else Oend+case when(instr(:ext_type,'M')&gt;0) then 4else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>	is TAB delimited, replace	
<pre>EXT_TYPE "case when(instr(:ext_type,'D')&gt;0) then 1else Oend+case when(instr(:ext_type,'F')&gt;0) then 2else Oend+case when(instr(:ext_type,'M')&gt;0) then 4else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>	the comma with \t.	
<pre>when(instr(:ext_type,'F')&gt;0) then 2else Oend+case when(instr(:ext_type,'M')&gt;0) then 4else Oend+case when(instr(:ext_type,'S')&gt;0) then 8else Oend+case</pre>		
<pre>when(instr(:ext_type,'M')&gt;0) then 4else 0end+case when(instr(:ext_type,'S')&gt;0) then 8else 0end+case</pre>		
when(instr(:ext_type,'S')>0) then 8else Oend+case		
)		
		when (inder (.exc_cype, v )>0) chen ideide dend
		1

## 3. Click Save.

For more information about importing Directory Listings, see "Import Sets" in the *Using the ETM*<sup>®</sup> *System* user guide or online Help.

**Versions Affected** ETM<sup>®</sup> System 4.1 and later.

Last Updated

3/3/2004

SecureLogix Corporation

13750 San Pedro, Suite 230 • San Antonio, Texas 78232 • (210) 402-9669 • www.securelogix.com

Support (877) SLC-4HELP • EMAIL support@securelogix.com • http://support.securelogix.com

ETM, TeleWall, TeleAudit, TeleView, TeleVPN, TeleIDS, TeleWatch Secure, TWSA, SecureLogix, SecureLogix Corporation, and the ETM, TeleWall, TeleAudit, TeleView, and TeleVPN Emblems and the SecureLogix Diamond Emblem are trademarks or registered trademarks of SecureLogix Corporation in the U.S.A. and other countries. All other trademarks mentioned herein are believed to be trademarks of their respective owners.

© Copyright 2004 SecureLogix Corporation. All Rights Reserved.

U.S. Patents No. US 6,249,575 B1, US 6,320,948 B1, US 6,542,592 B1, US 6,687,353 B1, and US 6,700,964 B1.

U.S. and Foreign Patents Pending.