

DSImport

eDirectory Import Tool

DSImport v1-1x



Dr. Wolfgang Schreiber

www.WSTools.de
www.WolfgangSchreiber.de



Copyright 2010 by Verlag Wolfgang Schreiber-Neumann
www.WolfgangSchreiber.de

Printed May 2010

All rights reserved. No part of this manual or software may be reproduced or modified without the prior written permission of the publisher.

Though all functions and descriptions of this software have been produced and tested carefully by the author, errors can never be completely excluded. Author and publisher cannot be made liable for any damage caused by incorrect usage, incomplete documentation, or software malfunction.

The publisher reserves the right to make changes to any part of the software or the documentation without obligation to notify any person or entity.

Publisher and author gratefully accept suggestions for future revisions of this product.

Table of Contents

| | | |
|-----------|--|-----------|
| 1 | OVERVIEW | 11 |
| 1.1 | WHY DSIMPORT? | 11 |
| 1.2 | LIMITATIONS AND COMPATIBILITY | 11 |
| 1.3 | RISKS AND PRECAUTIONS..... | 12 |
| 1.4 | NOTES AND TROUBLESHOOTING | 12 |
| 2 | QUICK START | 14 |
| 3 | THE DATA FORMAT | 15 |
| 3.1 | BASIC SYNTAX | 15 |
| 3.2 | USING QUOTES..... | 16 |
| 3.3 | USING DNSs VS. CNS..... | 16 |
| 3.4 | MULTI-TREE ENVIRONMENTS..... | 16 |
| 3.5 | MULTIVALUED ATTRIBUTES | 17 |
| 3.6 | FIELD SEPARATORS VS. VALUE SEPARATORS..... | 17 |
| 3.7 | WHITE SPACE | 18 |
| 3.8 | PSEUDO-ATTRIBUTES: PASSWORDS..... | 18 |
| 3.9 | IMPORT FILES FOR DELETING OBJECTS..... | 18 |
| 3.10 | USING AUXILIARY CLASSES | 18 |
| 4 | THE INPUT DATA..... | 20 |
| 4.1 | SELECTING THE FILE | 20 |
| 4.2 | SELECTING THE FILE FORMAT..... | 21 |
| 5 | THE GRID..... | 23 |
| 5.1 | OVERVIEW | 23 |
| 5.2 | REORDERING FIELDS/COLUMNS..... | 24 |
| 5.3 | DISABLING FIELDS/COLUMNS..... | 26 |
| 5.4 | THE CONTEXT MENU..... | 27 |
| 5.5 | EDITING DATA FIELDS | 28 |
| 5.6 | EDITING ATTRIBUTE NAMES IN THE HEADER ROW..... | 28 |
| 5.7 | DELETING FIELDS/COLUMNS OR ROWS | 28 |
| 5.8 | TESTING YOUR GRID | 28 |
| 6 | THE ACTION..... | 29 |
| 6.1 | DELETE OBJECTS..... | 29 |
| 6.2 | CREATE OBJECTS | 30 |
| 6.3 | MODIFY OBJECTS..... | 31 |
| a) | <i>Replace Values</i> | 31 |
| b) | <i>Add Values</i> | 31 |
| c) | <i>Remove Values</i> | 31 |
| 6.4 | COMPARE OBJECTS | 31 |
| 7 | LOGGING AND ERRORS | 32 |
| 8 | INSTALLATION | 33 |
| 9 | SUPPORT AND FEEDBACK | 34 |
| 10 | VERSION HISTORY, RECENT CHANGES | 35 |

1 Overview

1.1 Why DSImport?

DSImport is a tool that can be used to import external data from text or CSV files into eDirectory.

- **DSImport provides an easy way to update eDirectory from text files**
While it is possible to perform similar bulk import operations with LDIF files, the LDIF format is far from intuitive, prone to errors and requires significant text processing to get it into the right format.
CSV files are more intuitive and flexible.
- **DSImport handles a variety of text formats**
CSV strictly spoken stands for “Comma Separated Values”, but many implementations allow for or even require other formats (like semicolons) as value separators. Using DSImport you can dynamically choose the data format of your input file.
- **DSImport allows object creation, modification, deletion, and comparison**
You may create input files for DSImport that enable you to create new objects, modify objects by adding, replacing, or removing values; you may also use DSImport to compare your eDirectory objects with the data from your input file.

1.2 Limitations and Compatibility

- **eDirectory versions**

DSImport is compatible with NDS and eDirectory versions v4.x to v8.x running on any platform (NetWare v4.x and above, Linux, Windows, etc).

- **Operating system and client software**

To run DSImport you will need a 32bit Windows platform (e.g., Win9x, WinNT, or Win2k, WinXP, Win2k3, etc) and the Novell Client installed. The Microsoft client for Novell does not provide a full interface to eDirectory.

- **Memory requirements**

A minimum of 32MB RAM is recommended, though DSImport may run with even less RAM. In rare cases however, if you are working with huge import files, you may run out of RAM if the input file cannot completely be loaded into memory. In such infrequent cases, consider closing other applications, splitting the input file into smaller pieces, or adding RAM.

- **Disk space requirements**

Since DSImport itself does not need disk space except for the executable and the related DLLs, it will typically need less than 2MB of hard disk space and can be installed on a shared drive, e.g., on a NetWare server.

1.3 Risks and Precautions

- **Do not use DSImport thoughtlessly!**

DSImport is a tool that allows easy access and manipulation of eDirectory objects. The main objective during the development of DSImport was to allow fast modifications of eDirectory objects without additional user interaction, or confirmation.

DSImport is a powerful tool, capable of performing major eDirectory modifications. There is a risk of causing damage to your system when using DSImport without caution.

To protect you and your directory from those kinds of problems it is strongly recommended to regularly backup your eDirectory with NDS-aware utilities.

- **Can you break the rules?**

eDirectory offers server based security clearance. The server interprets all commands and requests from clients. eDirectory checks your privileges and rejects unauthorized requests. Whatever tool you use - it is not possible to circumvent this control. DSImport will not allow you to do more than what has been granted to you by eDirectory (i.e., the eDirectory Administrator).

- **Do your actions have to make sense?**

While eDirectory verifies your privileges and rejects unauthorized requests, eDirectory does not make a check for unwise operations. DSImport allows you to act towards eDirectory as thoughtful or brainless as you wish.

If you have Administrator rights and intend to use DSImport, you might wish to do so after your first flush of caffeine in the morning.

You might want to create an organizational unit in your existing tree (or create a separate tree) to test your actions before operating in the production environment.

1.4 Notes and Troubleshooting

- **ZEN objects**

DSImport works with eDirectory objects. Since some applications (especially NAL or ZENworks) store information outside of eDirectory, DSImport operations may succeed only partially.

To be on the save side, you should test your actions in a test environment before working with your production system.

- **Support or questions**

The fastest way to solve standard problems or questions with DSImport is to post an entry in the blog on <http://www.WolfgangSchreiber.de>

For problems that go beyond such support or that require engineering, you may contact wstools@WolfgangSchreiber.de - a minimum charge of \$50 may apply.

2 Quick Start

Importing data into eDirectory with DSImport is done in three easy steps:

- Create your input data file

For testing you may use a simple file with this content:

```
CN;Surname;Given Name
MParten;Parten;Michael
VRowe;Rowe;Valerie
```

- Start DSImport and select the target context (Tab: "Tree and Context")
Choose the container where you want to create the test objects
- Load the data file into DSImport, specifying the format of your data (Tab: "CSV -> eDirectory": "Load CSV")
Verify the displayed data in the grid
- Select the desired "Object Action" (like create, modify, or delete)
- Click "Start Add" to start the import
- Check the result log (Tab: "Results")

3 The Data Format

3.1 Basic Syntax

The Header line: The input data must contain a header row specifying the attribute names that you plan to import

The Data lines: the header line is followed by one or more data lines. The number of fields/attributes in the data lines must match the fields in the header line.

Comment lines: The data file may contain comment lines starting with a hash ('#') symbol

A very simple input file to create new user objects could look like this:

```
DN; Surname; Given Name; Telephone Number
.jdoe.myOu.myOrg; Doe; John; 123456
.mmouse.myOu.myOrg; Mouse; Micky; 654321
.mjackson.myOu.myOrg; Jackson; Michael; 123456
```

This sample used the semicolon “;” as field delimiter. Value delimiters or quote characters were not used.

Multivalued fields: you may import multiple values in one go. The multiple values need to be separated by a value separator of your choice, e.g. a comma “,”

```
DN; Surname; Given Name; Telephone Number
.mjackson.myOu.myOrg; Jackson; Michael, Jacko; 123456
```

If you forgot to tell DSImport what value separator you use, DSImport would treat the input “Michael, Jacko” as a single string, instead of treating it as two separate values.

When opening an input file, you may specify the field and value separator that you use. Therefore you could also use an input file like in the sample below, as long as you tell DSImport, that you’d rather use the comma “,” as field separator and the hash sign “#” as value separator:

```
DN, Surname, Given Name, Telephone Number
.mjackson.myOu.myOrg, Jackson#Michael, 123456
```

I think you get the picture.

When loading an input file, DSImport allows you to declare the field and value separators and displays the input data in a grid, where each column represents one attribute.

Scrutinize the grid display to verify that DSImport has taken care of field and value separators correctly.

3.2 Using Quotes

There may be cases, where the character that you use as field or value separator needs to occur in the value.

Let's assume that you use commas as field separator, and a comma needs to be in one of the values. In this case, you can embed the value in quotes. Example

```
CN, Surname, Given Name, Description
mjackson, Jackson, Michael, "Michael, King of Pop"
```

The quotes are used to make sure that embedded separators are not interpreted as field separators.

Whenever field values contain the specified field or value characters, use quoted around the values.

DSImport may be told to use single quotes (') or double quotes (") as quote characters

3.3 Using DNs vs. CNs

The first column in the input data is used to identify the target object. Typically, you will use the target object name in one of two forms: the relative name (CN, e.g. "JDoe") or the fully qualified name (DN, e.g. ".JDoe.myOrgUnit.myOrg"). If you wish, you may also use partially relative contexts – see the Novell eDirectory documentation on contexts and naming (<http://www.novell.com/documentation/edir88/edir88/data/fbacgbeb.html>) for details.

CN/Relative names: when using relative names you need to tell DSImport the context, where the object is (or needs to be) located. This is done with the tree view selector: select the appropriate tree and context where the objects are or should be located, then proceed with processing the data.

DN/Fully qualified names: When using fully qualified object names, the DN typically starts with a leading period to resolve the name from the top of the tree, no matter where the current application context is set.

Typically, names are either specified as relative CNs, or as fully qualified DNs with a leading period.

You may even combine these two approaches in the same input file: entries with relative names will be positioned in the selected context, while entries with a fully qualified name will be positioned in that specified context

```
CN;Surname;Given Name
Test1;Test SN;Test GN
.Test2.myOu2.myOrg;Test SN;Test GN
.Test3.myOu3.myOrg;Test SN;Test GN
```

The header row label for the first column will be ignored in this case and can be either "CN" or "DN".

3.4 Multi-Tree Environments

DSImport is able to handle client connections to multiple trees, but obviously you will need to let DSImport know, what tree to talk to when processing objects. Therefore you will need to select the target tree with the tree view selector.

3.5 Multivalued attributes

DSImport can handle multivalued attributes. These can occur in 3 scenarios:

- a) you want to add multiple attribute values to the same object attribute, keeping the original values
- b) you want to add multiple attribute values to the same object attribute, overwriting the original values
- c) you want to remove multiple attribute values from the same object attribute

The only thing that you need to take care of is that the individual values that you plan to add or remove are separated with the correct value separator. You may for example use the semicolon as field separator and the comma as value separator.

Assume you want to add 3 telephone numbers to an account; you could use an input file like this

```
CN; Telephone Number; Title
JDoe; 123, 456, 789; Manager
```

3.6 Field Separators vs. Value Separators

Field separators are needed, whenever you handle one or more attributes in your input file. Since the only situation where no attribute is needed is when deleting objects, this means that you always need to declare and use field separators.

Typically the field separator is a comma or semicolon, but DSImport allows you to use a couple of other characters as well.

Every time you want to use more than one value for the same field (e.g. use 2 telephone numbers), you will need to separate these values with a value separator of your choice.

Assume you want to add 3 telephone numbers to an account; you could use an input file like this

```
CN; Telephone Number; Title
JDoe; 123, 456, 789; Manager
```

In this example, the field separator is the semicolon, while the value separator is a comma.

Alternatively you could use an input file like this

```
CN, Telephone Number, Title
JDoe, 123; 456; 789, Manager
```

In this example, the field separator is the comma, while the value separator is a semicolon.

Obviously, you need to declare different characters as field and value separators, since otherwise DSImport couldn't identify the values correctly.

3.7 White Space

As you may have noticed, there was white space (blanks) in the sample input files. As a rule of thumb you can say that white leading or trailing white space around values will be removed during the input processing.

However, if you – however unlikely this is - need to have the leading or trailing white space in the attribute value, embrace the value with quotes.

```
CN, Title  
JDoe, "Manager "
```

You will also need to use quotes, if your values contain one of the field or value separators

```
CN, Title  
JDoe, "Guru, Saint, Great Boss"
```

3.8 Pseudo-Attributes: Passwords

eDirectory does not store passwords in regular attributes - you cannot set/change/read passwords like any other attribute.

To allow you to set objects' passwords during normal object creation and modification, DSImport uses a pseudo-attribute 'password' that you can simply add as a column in your input file.

While processing the input file, DSImport will handle this pseudo-attribute.

This little sample could create a user and assign the password 'secret'

```
CN, Surname, Given Name, password  
JDoe, Doe, John, secret
```

3.9 Import Files for Deleting Objects

You may use import files with DSImport to delete existing objects. The only required information here is the first data column, i.e., the relative CN or the fully qualified DN.

```
CN  
Mmouse  
Dduck  
.jdoe.acme.org
```

The input file may contain additional columns, but these will be ignored when processing the input data.

However, if you plan to your such input files with extra data columns, be aware that object lines may be ignored if they do not match the field specification from the header row; after all, DSImport does not know what operation you wish to perform when loading an input file, so it will make sure that data lines match the definition in the header row.

3.10 Using Auxiliary Classes

You may be using custom auxiliary attributes¹ in your tree, and you want to create new objects with such auxiliary attributes, or to add such auxiliary attributes to existing users.

To do so, you will also have to add the auxiliary class to the object before adding that custom attribute.

DSImport allows you to add auxiliary classes to the objects while doing the create/modify operation. Simply select one of the auxiliary classes from the dropdown list to have it added automatically.

Adding more than one auxiliary class: In some infrequent cases, you may have to add more than one auxiliary class when processing the input data. The current DSImport version v1-1 allows you to add no more than one auxiliary class per import.

¹ Actually, I should use the terms more specific: eDirectory does not know the term “auxiliary attribute”. There is the concept of the “auxiliary class” which can be added to existing objects; this allows you to add attributes to the objects that have not been declared in the object’s base class, but only in the added auxiliary class.

When talking about “auxiliary attributes” I refer to such attributes which are part of such auxiliary classes.

4 The Input Data

4.1 Selecting the File

Click the “Load CSV” button to open the file selection box and browse to the file that you would like to import.

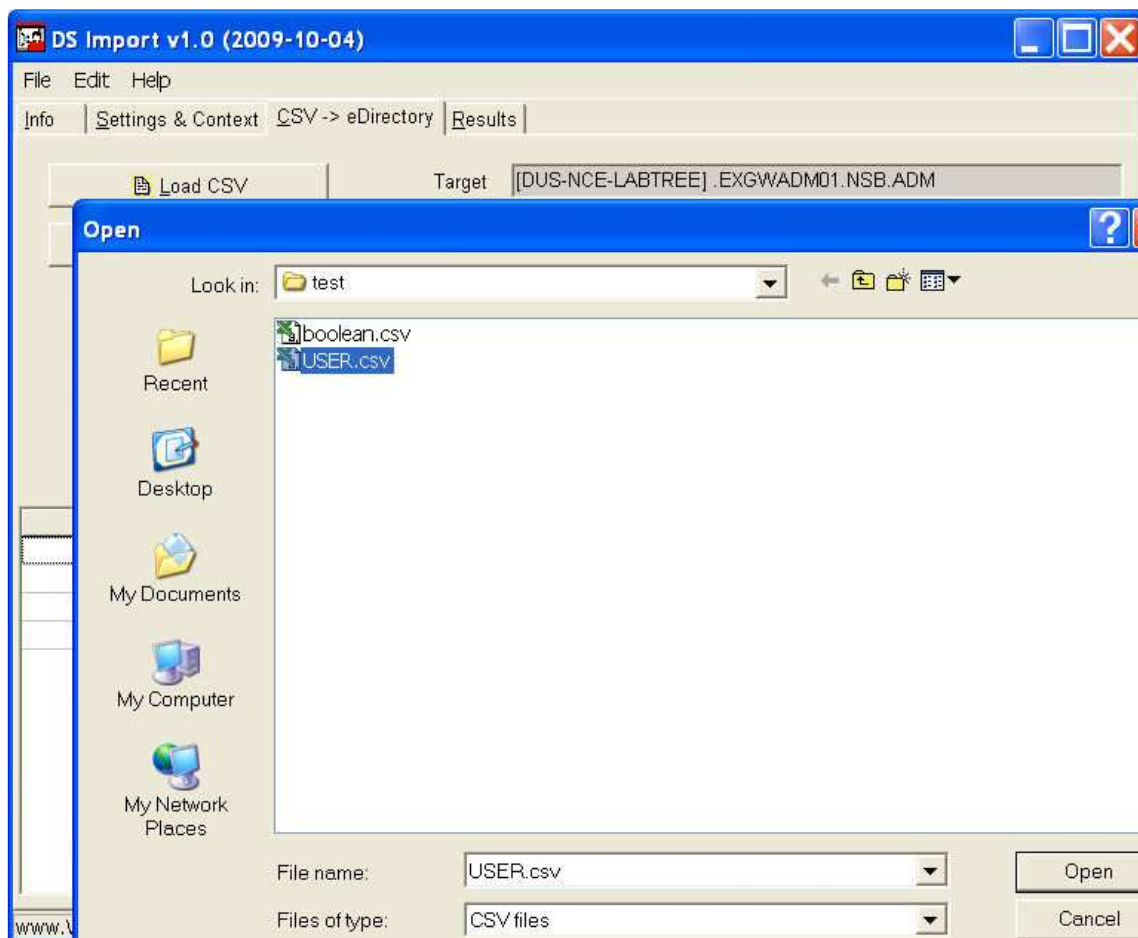


Figure 1: Selecting the Input Data

4.2 Selecting the File Format

After selecting the file, you'll receive a dialog that allows you to tell DSImport about the file format.

The preview grid gives you an impression if your choice of delimiters and quote characters was appropriate. The preview will display the first lines of the input data in the preview grid.

If the input data contain date or time attributes (e.g., "Login Expiration Time"), you will also need to specify the data/time format that your input file uses to represent these data.

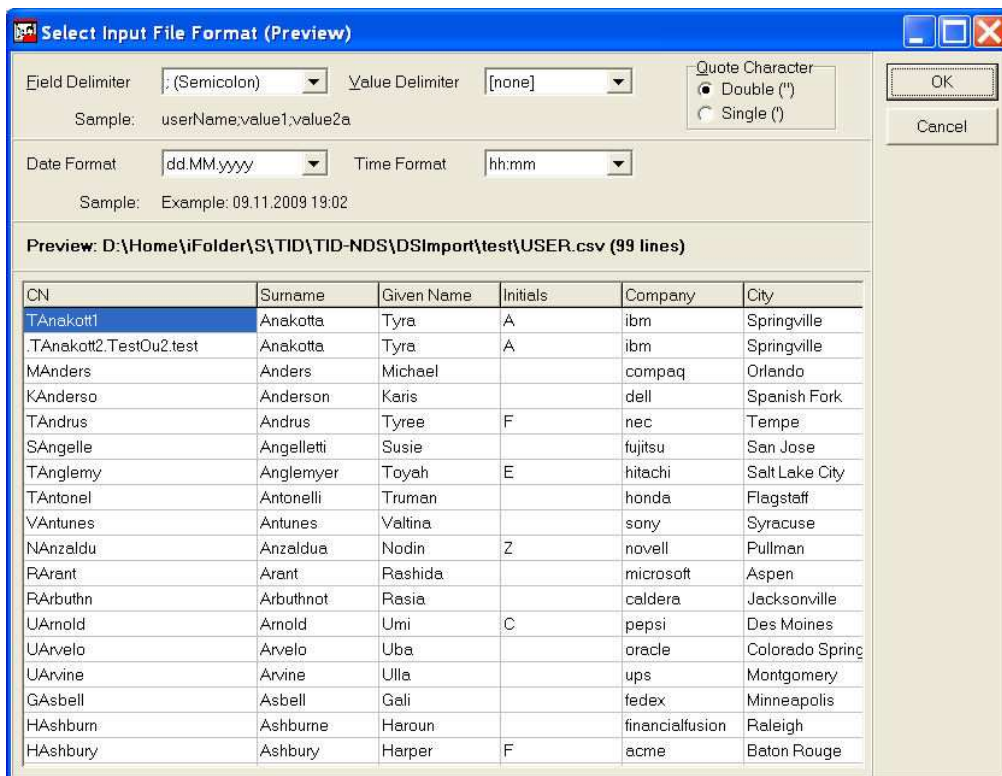


Figure 2: Preview with Correct Delimiter Selection

The dialog allows you to select the field and value delimiters that correctly represent your file format, as well as select the quote characters and date/time formats

The preview will dynamically reflect your choice of delimiters and quotes, thereby giving you an indication if your selection was correct.

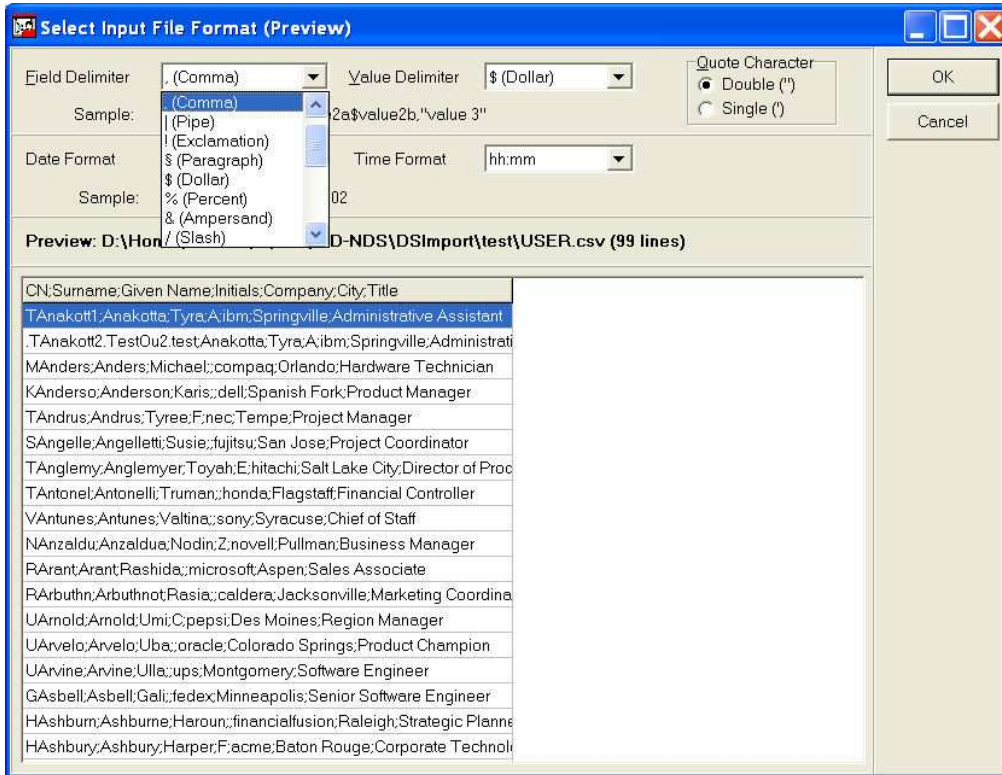


Figure 3: Preview with Incorrect Delimiter Selection

After choosing the input format, press "OK"

5 The Grid

Once you have selected the input data file and the data file format, the complete file will be loaded into the data grid.

If input lines did not match the file format specifications, e.g. because they contain an incorrect number of fields, you will find some details about the skipped lines in the report window.

Important: While the grid allows you to edit the contents and structure of the displayed data, your original input data file will not change. All modifications of grid data take place in memory and will be discarded as soon as you close DSImport or load the next input file. You may, however, save the contents of a modified grid to a new file.

5.1 Overview

The data grid will display your input data in table format, each row representing a valid line from your input file, each column representing a field/attribute.

Comment lines (starting with '#') will not be displayed in the grid.

Invalid lines also will not be displayed.

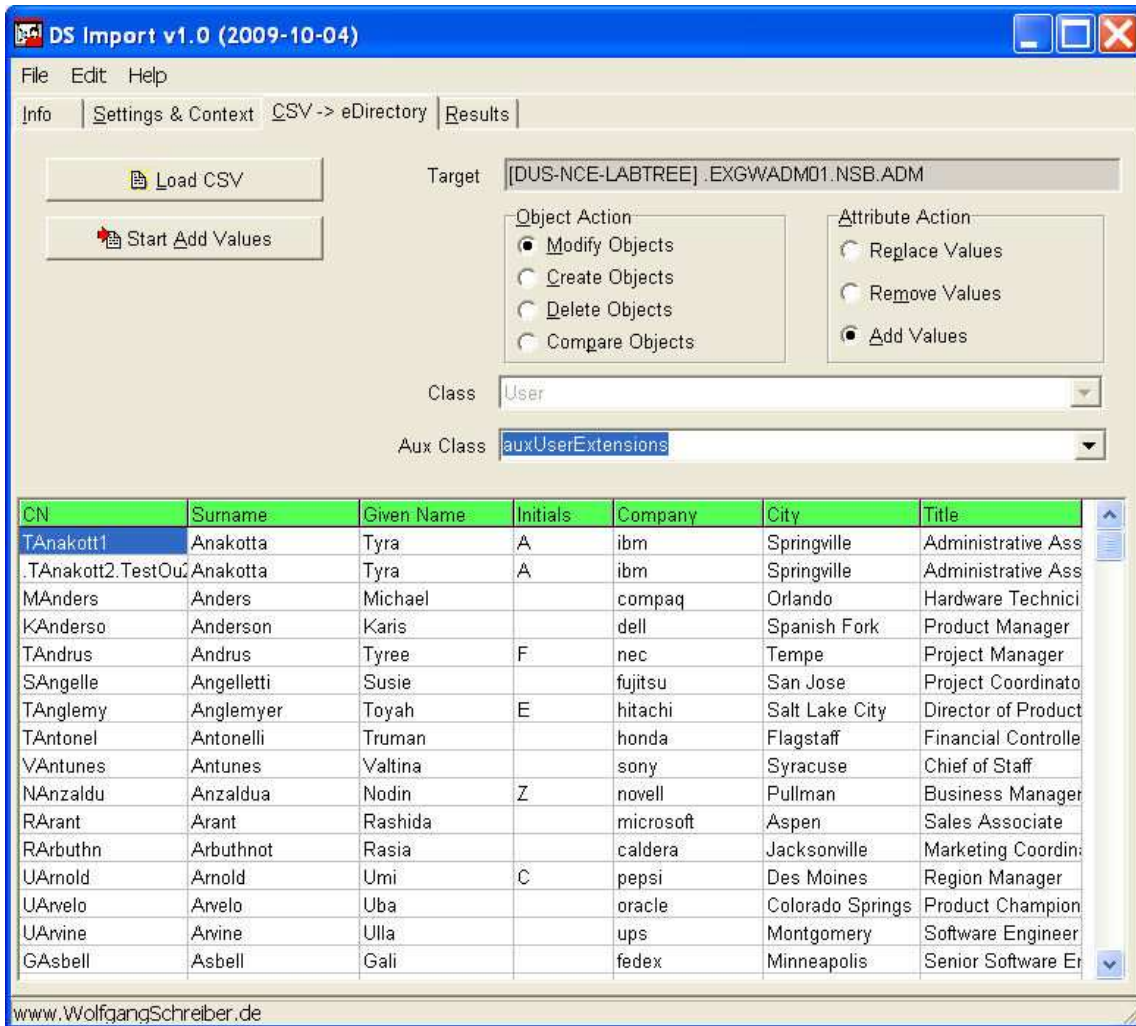


Figure 4: The Grid

If you feel the grid represents your data correctly, you can process the data right now, but let us first take a look at some of the other options that you have in the grid.

5.2 Reordering Fields/Columns

You may click on a header field and drag it to a different location to reorder the columns.

However, since typically it doesn't make a difference if one attribute is modified before or after another, there is only one major reason to do so:

The first column represents the object that you want to modify, and your data may contain the target object in a different column.

Example: You have an input file with 2 columns: "Manager" and "Assistant". During the first run, you may have modified the "Assistant" attribute for all the objects from the first column.

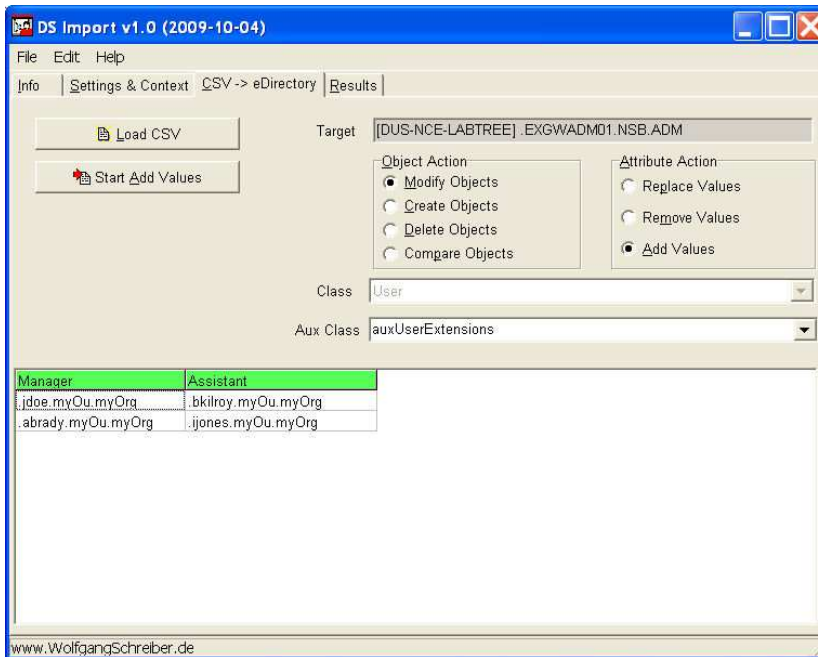


Figure 5: Moving Columns - before

Then, you move the second column to the first position, and now modify the “Manager” attribute for all objects in the “Assistant” column.

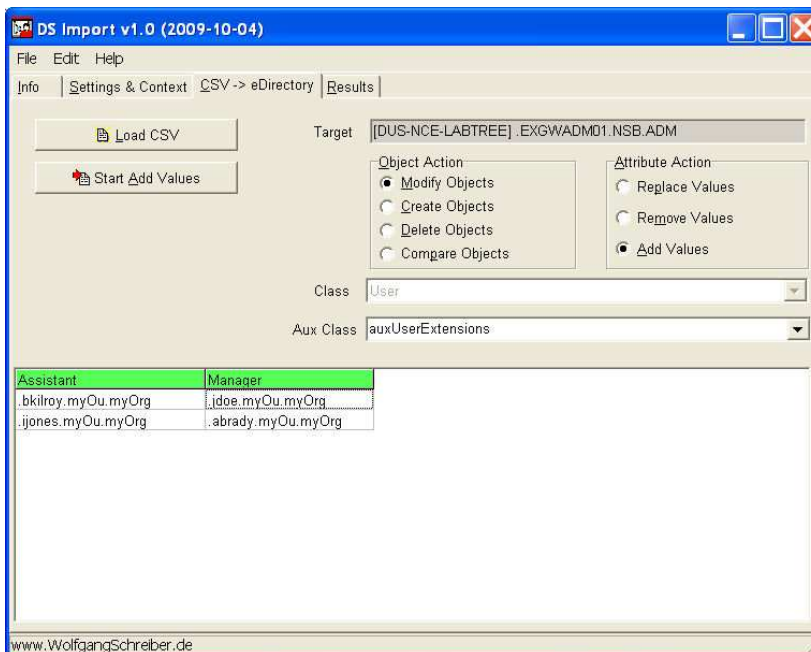


Figure 6: Moving Columns - after

Using the same approach, you could also modify other back link relationships like group memberships.

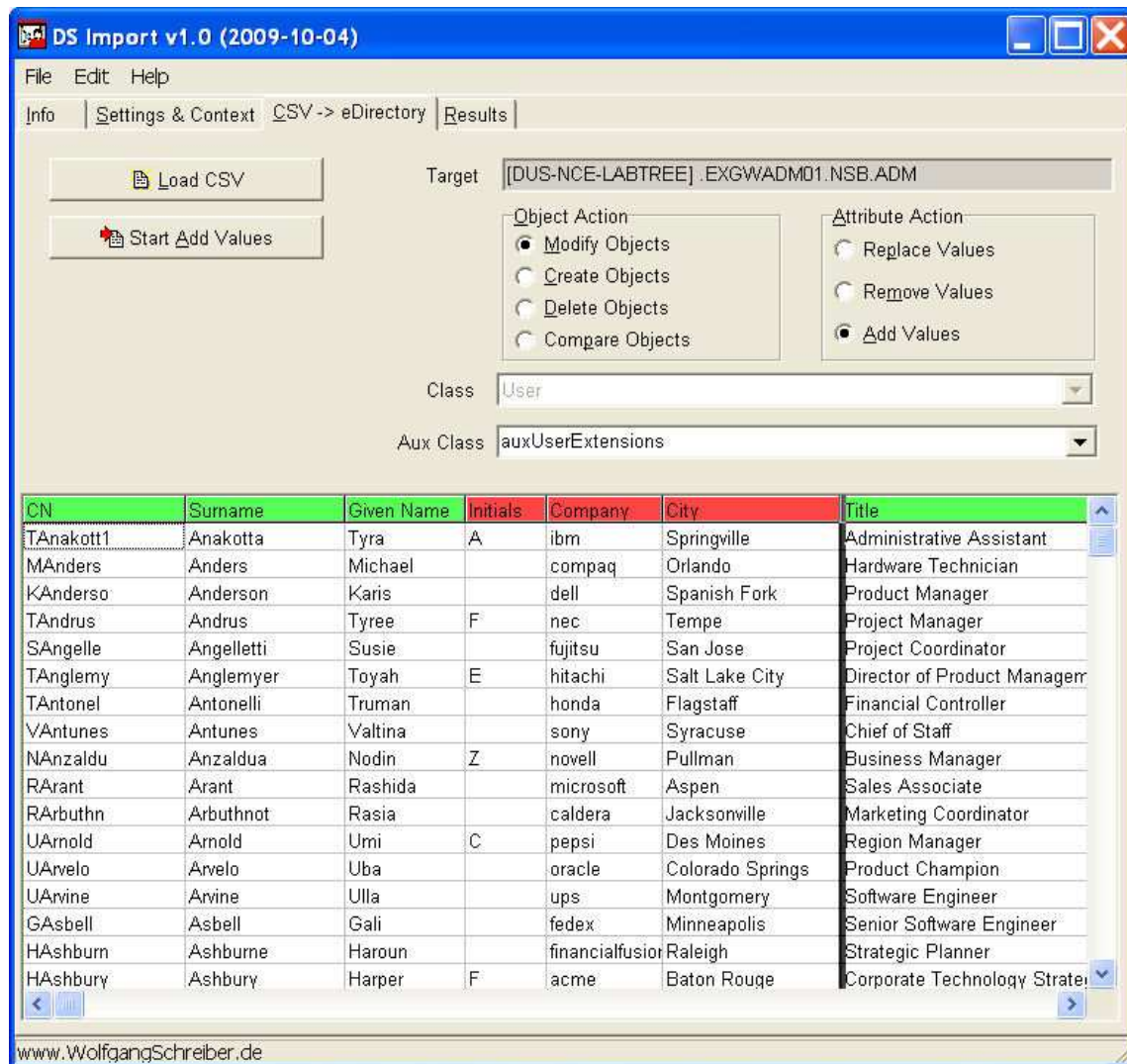
Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.3 Disabling Fields/Columns

You may have an input file with multiple fields/attributes, but for some reason you may want to exclude selected attributes from being imported into eDirectory.

While one option may be to edit your input file accordingly, you can much more easily exclude columns by deactivating them or by deleting the column from the grid (see 5.7 Deleting Fields/Columns)

Deactivate or reactivate columns by clicking on the field name in the header row.



Deactivated columns are displayed in red color, active columns are displayed green.

Deactivated columns will be skipped when processing the data.

Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.4 The Context Menu

When right-clicking in the grid you will get a popup context menu with some grid/cell/row/column specific options.

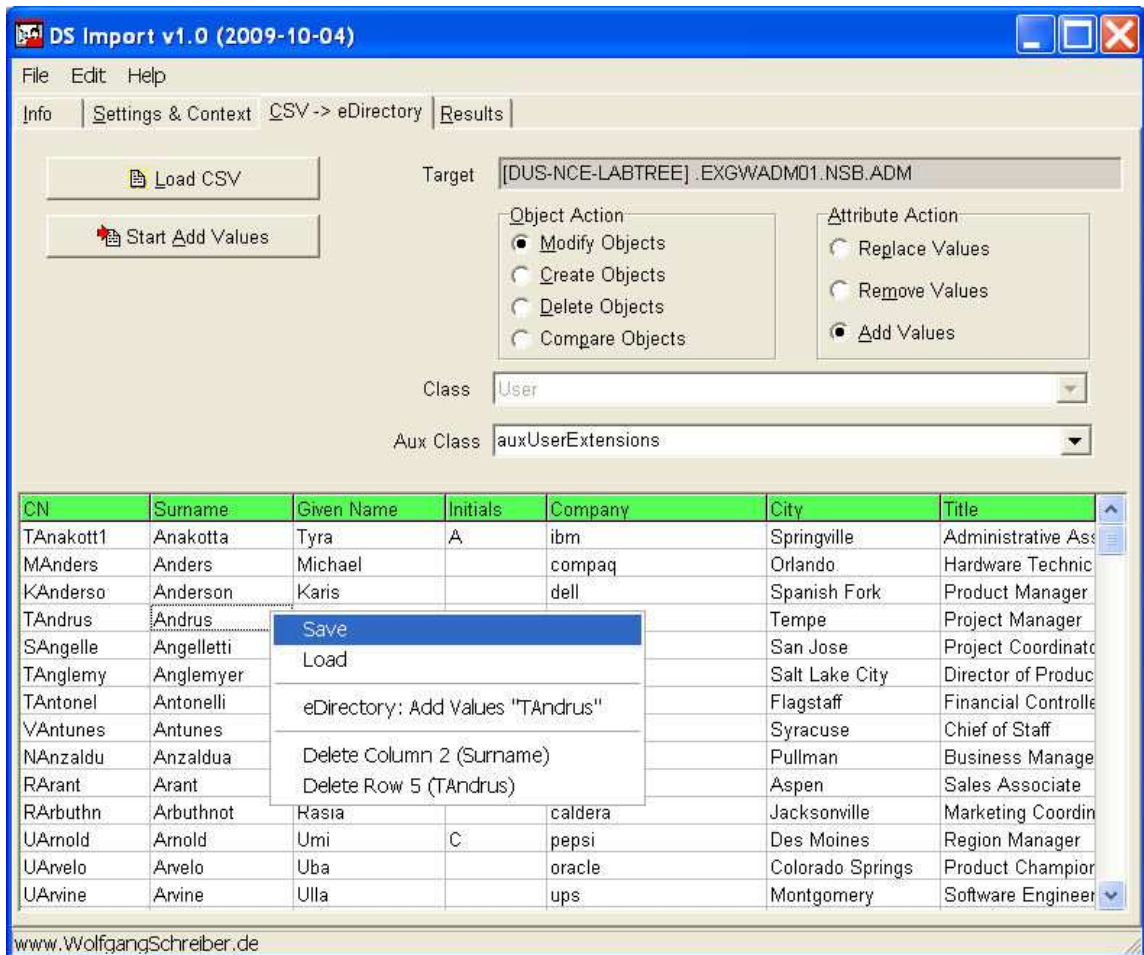


Figure 7: The Grid Context Menu

The context menu allows you to ...

- Save the current contents of the edited grid to a new file
- Load a new input file
- Perform the selected action (create/modify/delete) for a single object
- Delete the selected column from the grid
- Delete the selected row from the grid
-

Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.5 Editing Data Fields

There may be cases, where the attribute values need to be modified. Click twice into the cell that you want to edit, and then enter your correction.

Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.6 Editing Attribute Names in the Header Row

There may be cases, where the attribute names in the header row need to be modified. By default, the header row is protected and set to read-only. To remove this protection, right-click in the header row and make it editable.

Now you may click into the header labels and change the attribute names.

Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.7 Deleting Fields/Columns or Rows

You may decide to not only deactivate selected columns, but completely remove the column from the grid.

Right-click somewhere into the selected column and choose the menu option to delete the column from the grid.

The equivalent action may be performed to remove selected rows from the grid.

Note: changes take place in grid memory, not in the original data file. To make the changes permanent, use the context menu and save the grid to a new file.

5.8 Testing your Grid

Before importing the whole data grid into eDirectory, you may want to test the selected action with one or more selected objects.

You may easily do so by right-clicking onto the object line, and selecting the eDirectory action. The context menu text will reflect the action that you have chosen in the action selector.

6 The Action

The current version of DSImport supports four main action types: you may use the tool to **delete**, **create**, or **modify** objects, and you may use it to **compare** existing objects with data from the grid.

6.1 Delete Objects

The Delete action is the simplest one, because the only required input is the CN or DN of the target object.

The input file may contain more columns, but only the first column will be considered during the action.

The form input areas for object "Class", "Auxiliary Class", and "Attribute Action" are disabled, since they are irrelevant for object deletion.

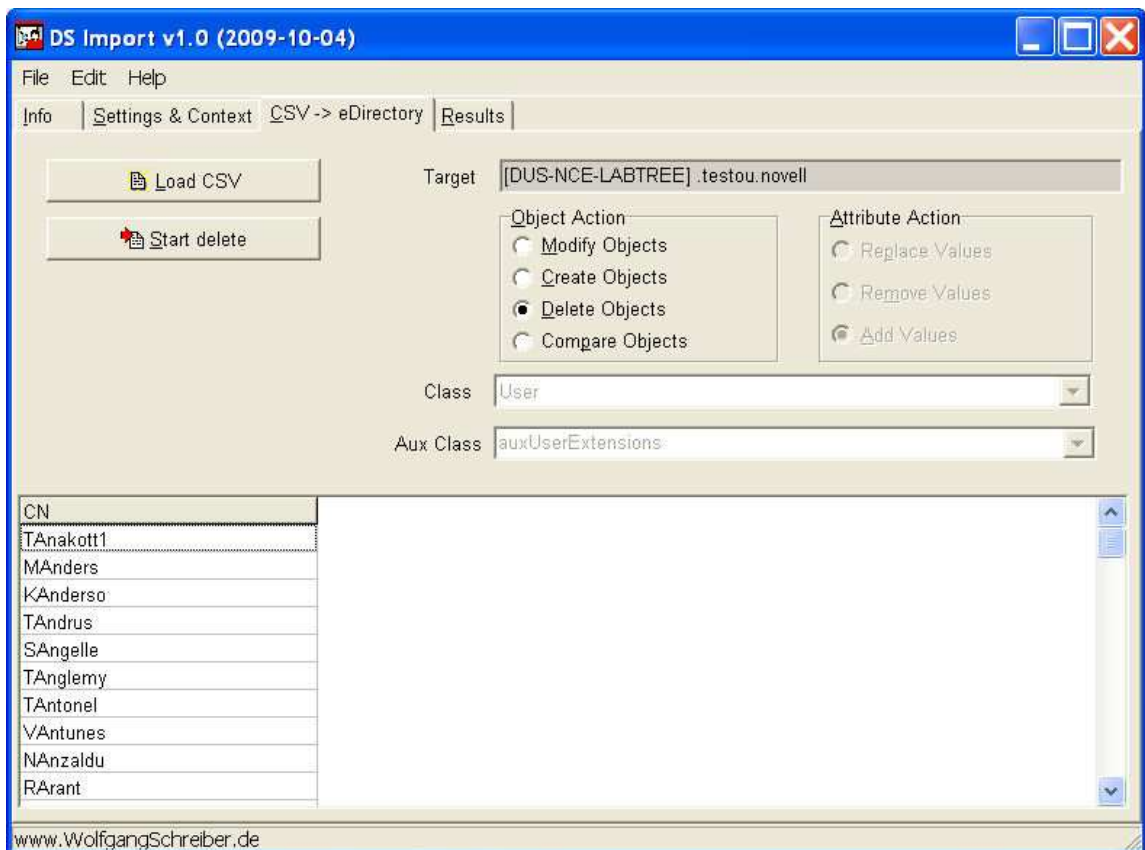


Figure 8: Object Action - Delete

Note however, that DSImport will do a sanity check on the whole input file; if your file contains more than one column, the number of data fields for each object must match the fields in the header row. After all DSImport read the file before you tell it what action you'd like to perform with the file.

6.2 Create Objects

When creating new objects, you will need to use the dropdown list to specify the object class.

Besides this, the input file needs to contain all mandatory attributes that are required for the selected object class.

If you are not sure what attributes are mandatory for the selected class, check the hint when you move the mouse over the selected class in the dropdown list.

The attribute "Object Class" does not need to be in the input file, since the class is selected from the dropdown list.

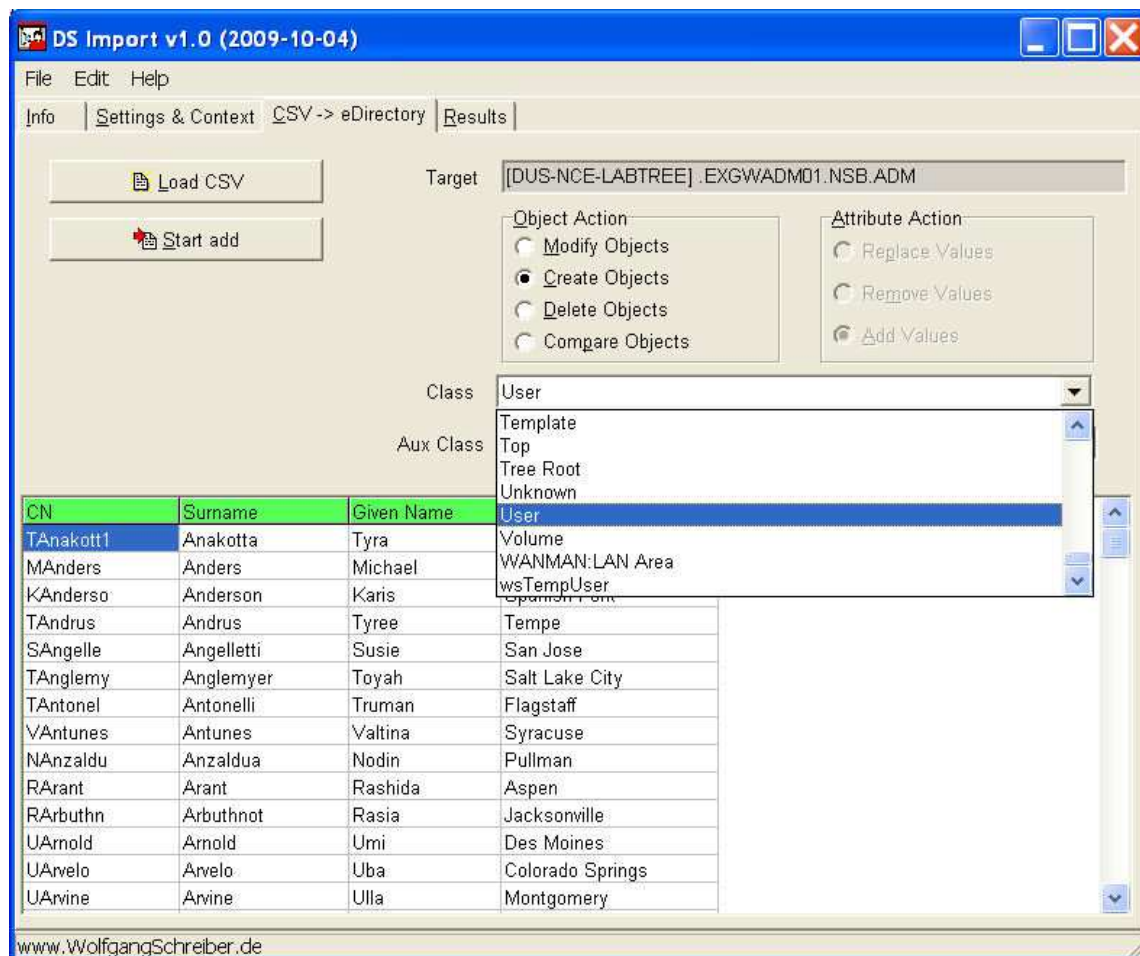


Figure 9: Object Action - Create Objects

Optionally, you may also add an auxiliary class from the second drop down list.

The form input area "Attribute Action" is disabled, since it is irrelevant for object creation. To assign initial passwords, use the pseudo-attribute 'password'.

6.3 Modify Objects

When modifying objects from an input file, multivalued attributes require special attention, since you could want to **a) overwrite** existing values with the imported ones, **b) add** the imported values to the existing ones, or **c) remove** selected values from the object.

Therefore, when selecting the object action “Modify Objects” you will also need to specify, what you would like to do with the imported values.

a) Replace Values

Remove any existing values and set the attribute to the new data from the input file.
To change users' passwords, use the pseudo-attribute 'password'.

b) Add Values

Keep existing values and add the value(s) from the input file

c) Remove Values

Remove the values from the input file if they are present on the target object.

6.4 Compare Objects

You may use DSImport to compare the objects' actual attributes with the attribute values stored in the grid. DSImport will read the selected attributes from eDirectory, and then compare the results with the data in the grid.

This feature may also be useful after a “Create” or “Modify” action; you may use the same grid that you used to perform the action to validate the results.

7 Logging and Errors

For most eDirectory actions and for all the errors that occurred during the operation, DSImport will generate a log entry.

Use this logging feature to check the results after a bulk operation to see if any warnings or errors occurred.

You may use the context menu in the logger window to clear the result screen or to export the log to a file.

8 Installation

There is no installation required. Simply copy all files from the distribution media to a directory of your choice.

To remove DSImport, simply delete the files.

In case you have received DSImport with a Setup application, you will find an uninstall program after doing the installation.

9 Support and Feedback

DSImport v1-1 is distributed as Shareware and not supported.

You may post questions or feedback using the respective links on www.wolfgangschreiber.de or www.wstools.de

10 Version History, Recent Changes

| DSImport | | |
|----------|-----|---|
| 2009.10 | 1.0 | First release |
| 2010.05 | 1.1 | Added password support for object creation/modification |
| | | |