End User License Agreement

END-USER LICENSE AGREEMENT FOR ToolSS ("SOFTWARE PRODUCT"). IMPORTANT PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE CONTINUING WITH THIS PROGRAM INSTALL: SS Solutions & Consulting LLC End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) for the SS Solutions & Consulting LLC software product(s) identified above which may include associated software components, media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. This license agreement represents the entire agreement concerning the program between you and SS Solutions & Consulting LLC, (referred to as "licenser"), and it supersedes any prior proposal, representation, or understanding between the parties. **If you do not agree to the terms of this EULA, do not install or use the SOFTWARE PRODUCT.**

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE.
The SOFTWARE PRODUCT is licensed as follows:
(a) Installation and Use.
SS Solutions & Consulting LLC grants you the right to install and use copies of the SOFTWARE PRODUCT on your computer running a validly licensed copy of the system for which the SOFTWARE PRODUCT was designed [e.g., Windows 95, Windows NT, Windows 98, Windows 2000, Windows 2003, Windows XP, Windows ME, Windows Vista, EMC Documentum]. All other brands and product names are trademarks of their respective owners.
(b) Backup Copies.
You may also make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.

2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.
(a) Maintenance of Copyright Notices.
You must not remove or alter any copyright notices on any and all copies of the SOFTWARE PRODUCT.
(b) Distribution.
You may not distribute registered copies of the SOFTWARE PRODUCT to third parties. Evaluation versions available for download from SS Solutions & Consulting LLC 's websites may be freely distributed.
(c) Prohibition on Reverse Engineering, Decompilation, and Disassembly.
You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
(d) Rental.
You may not rent, lease, or lend the SOFTWARE PRODUCT.
(e) Support Services.
SS Solutions & Consulting LLC may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA.
(f) Compliance with Applicable Laws.
You must comply with all applicable laws regarding use of the SOFTWARE PRODUCT.

3. TERMINATION
Without prejudice to any other rights, SS Solutions & Consulting LLC may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT in your possession.

4. COPYRIGHT
All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by SS Solutions & Consulting LLC or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by SS Solutions & Consulting LLC.

5. NO WARRANTIES
SS Solutions & Consulting LLC expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT is provided 'As Is' without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, noninfringement, or fitness of a particular purpose. SS Solutions & Consulting LLC does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the SOFTWARE PRODUCT. SS Solutions & Consulting LLC makes no warranties respecting any harm that may be caused by the transmission of a computer virus, worm, time bomb, logic bomb, or other such computer program. SS Solutions & Consulting LLC further expressly disclaims any warranty or representation to Authorized Users or to any third party.

6. LIMITATION OF LIABILITY
In no event shall SS Solutions & Consulting LLC be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if SS Solutions & Consulting LLC has been advised of the possibility of such damages. In no event will SS Solutions & Consulting LLC be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. SS Solutions & Consulting LLC shall have no liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information.

SS Solutions & Consulting LLC
ToolSS
CreateGroups
Installation and User Guide

End User License Agreement 1
Overview....................2
Background....................3
Utility Features and Limitations 3
Deployment....................4
Spreasheet Template Usage 5
Execution 6
Spreadsheet Mode....................6
Command Line Mode 7
File Preparation.................... 7
Command Line Execution 8
Other Processing Notes....................10
Known Issues....................11
Single Quotes....................11

Overview

The CreateGroups utility was developed to provide Documentum analysts, administrators, or developers with automation for the design and deployment of group structure within a Documentum repository. The CreateGroups utility was developed as a Java class for easy execution as a command line utility, leveraging the Java classes (DFC) provided with Documentum. The CreateGroups class accepts a CSV input file that can be generated using the provided *CreateGroups-Template.xls* spreadsheet. Designers can use the input file template to define the group structure to meet requirements. Once the specifications are complete, CreateGroups offers two modes for creation:

1. Spreadsheet Mode: The spreadsheet template includes a worksheet with macros that will allow execution of the CreateGroups class from any workstation where JRE, DFC, and CreateGroups are installed.
2. Command Line Mode: A CSV file can quickly and easily be generated from the input file template. This CSV file is then used as input for the CreateGroups class run in command line mode from any workstation where JRE, DFC, and CreateGroups are installed.

The CreateGroups utility processes each record within the input file and creates a dm_group object of the defined group class (if group does not exist) and assigns the group based on the defined group structure (if assignment does not already exist). A log is generated for each execution.

Background

The generation of group structure is a common requirement for repository design and configuration. The CreateGroups template provides and easy way to define the group structure design and can be used during requirements gathering and design workshops to finalize the desired structure.

Utility Features and Limitations

The following provides a brief summary of the functionality for the utility:

1. CreateGroups template spreadsheet provides for input of group structure in hierarchical format. For each defined group, spreadsheet provides for input of group_class (required; typically group, but other classes such as role are permissible).
2. The CreateGroups utility currently supports a maximum of ten group levels below the root group level.
3. Group names must conform to system specifications. The CreateGroups utility does not enforce these explicitly. If group names do not conform to system specification, an exception will be thrown. Note: all group names in Documentum are forced to all lower case. Even if you specify mixed case, they will be created as lower case when saved. Always use lower case for all group names.
4. Group names cannot end in a trailing space character. Although DCTM clients will allow this, it appears to be a DFC limitation: [DM_GROUP_E_TRAILING_SPACE_IN_NAME]error: The group name 'Approvals ' contains trailing space(s)
5. If the utility encounters a group that already exists, it skips creation of that group. Note: it does not verify or update other Properties for this case. This also allows for reprocessing of the input file if any exceptions are encountered, once the exceptions are resolved.
6. If the utility encounters a group that whose assignment already exists, it skips assignment of that group. This also allows for reprocessing of the input file for processing group hierarchy if groups already exist.
7. The utility will never delete groups; it will only add groups or skip groups if they already exist. The utility will never remove assignments; it will only add assignments or skip if assignment already exists.
8. If an exception is encountered for a group, then an error is posted to the log. All subsequent subgroups of that group will also fail. In such cases, it is necessary to correct the exception and re-run the input file.
9. The utility does not currently include support for Globally Managed groups. Default values will be assigned for this Property. The utility has ***not*** been tested in Federations. **Use in repositories associated with Federation should not be performed without prior testing**.

10. Command line Java utility that will accept parameters for username, password, repository name, and input file.
11. The input file must be in CSV format (not XLS). It is necessary to use Save As from Excel to convert to CSV format.
12. The input file must include the header row; this is skipped during processing.
13. Utility can be run from any workstation with JRE and DFC installed. The CreateGroups JAR is compiled for both Java 1.4 (DCTM 5.3) and 1.5 (DCTM 6.0, 6.5) compatibility.
14. Establish a session for the user in the repository as supplied by command line parameter. The user is expected to have at least GROUP privilege in the system.
15. Process the input file to create groups, assign metadata, and perform assignment to create defined group structure.
16. No additional or special handling will be performed for audit trail behavior; standard, default behavior will occur during utility execution.
17. Data Dictionary attribute validation is performed for each created group. If validation fails, the group is not created. Note: Typically, there are not attribute validation rules for dm_group, but if custom validation rules are added, they will be enforced.
18. An execution log is created for each execution with a unique file name within the directory from which the utility was executed. The log will minimally include the unique r_object_id and group path for each object created. Exception messages are posted for records with errors. This report may be used for historical reference.

Deployment

A deployment package consisting of a ZIP archive has been provided that contains the following:

Name	**Description**
CreateGroups-Template.xls	Input spreadsheet template.
jar	Directory containing Java Archives for deployment to machine where it is desired to run the CreateGroups utility. Note: There are two compiles provided, Java 1.4 and Java 1.5.
Install and Admin Guide	This document.
ReleaseNotes	Running log of revisions.
Sample	Directory containing sample spreadsheet, CSV, and output.

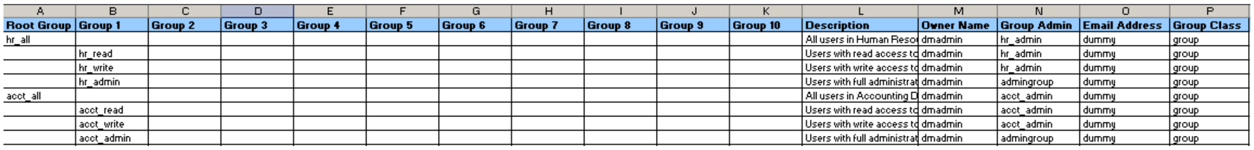
The CreateGroups can be deployed to any machine that has Java JRE (or SDK) and Documentum DFC installed by copying the CreateGroups.jar file to a directory of choice. This could be done on the Content Server, but this is not necessary.

Spreadsheet Template Usage

The *CreateGroups-Template.xls* is a spreadsheet that contains pre-formatted input for the design of group structure and assignment of object type and permissions.

Tip: It is a good practice to save a backup of the CreateGroups-Template.xls file in a safe location. Additionally, it is a good habit to always use File > Save As immediately after opening the template to save with a new file name. These practices will avoid inadvertent overwriting of the original template file.

The layout of the spreadsheet cannot be modified from its original layout, as the utility is expecting the defined order within the input file. The following shows a simple example:



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Root Group	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**	**Group 8**	**Group 9**	**Group 10**	**Description**	**Owner Name**	**Group Admin**	**Email Address**	**Group Class**
hr_all											All users in Human Reso	dmadmin	hr_admin	dummy	group
	hr_read										Users with read access to	dmadmin	hr_admin	dummy	group
	hr_write										Users with write access to	dmadmin	hr_admin	dummy	group
	hr_admin										Users with full administrat	dmadmin	admingroup	dummy	group
acct_all											All users in Accounting D	dmadmin	acct_admin	dummy	group
	acct_read										Users with read access to	dmadmin	acct_admin	dummy	group
	acct_write										Users with write access to	dmadmin	acct_admin	dummy	group
	acct_admin										Users with full administrat	dmadmin	admingroup	dummy	group

Each column within the spreadsheet represents a group level. This provides a fairly good representation of the nested group structure. In some cases, it may be desirable to size the columns for improved readability.

Each line within the spreadsheet represents a single group, and therefore a group name can only exist in one column per line. Put another way, no line within the spreadsheet should have more than one group or cabinet name defined.

Each line within the spreadsheet *must* have an Group Class (group_class) value, which must contain one of the permissible values (see *Documentum Object Reference* for permissible values). The specified Object Type name must *exactly* match the permissible values. The Config&Run worksheet contains a list of permissible values, but these should always be confirmed with vendor documentation. If no Group Class is specified, an exception will be thrown for that record.

If no Owner Name (owner_name) value is provided, the system will revert to default assignment of Owner Name.

Description (description) and Group Admin (group_admin) are optional.

Email Address (group_address) is required. The example above shows a workaround to supply a dummy value if no email address is desired.

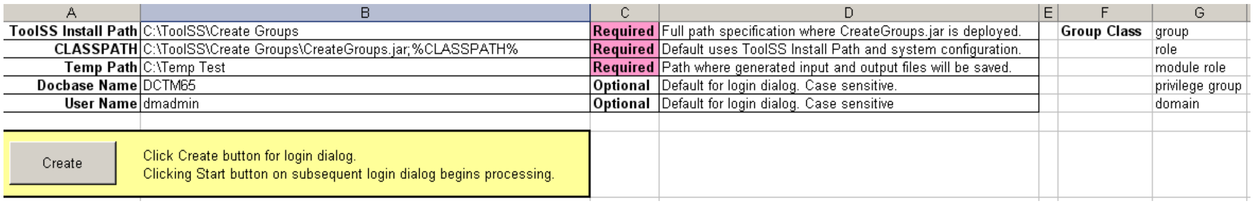
Owner Name and Group Admin must contain valid user or group names that either exist or are created during execution. The utility does not validate this, and the system does not appear to enforce this on save. Be sure to use valid user or groups names for these fields. Assignment of user/group names that do not exist can lead to data integrity issues within the repository.

Execution

Once group input data is complete, CreateGroups provides two modes for execution: Spreadsheet Mode and Command Line Mode.

Spreadsheet Mode

In spreadsheet mode, the CreateGroups class can be invoked directly from the spreadsheet on any workstation where JRE, DFC, and CreateGroups are installed. Once the group configuration input is completed, CreateGroups can be invoked from the Config&Run worksheet. A sample of the Config&Run worksheet is provided below:



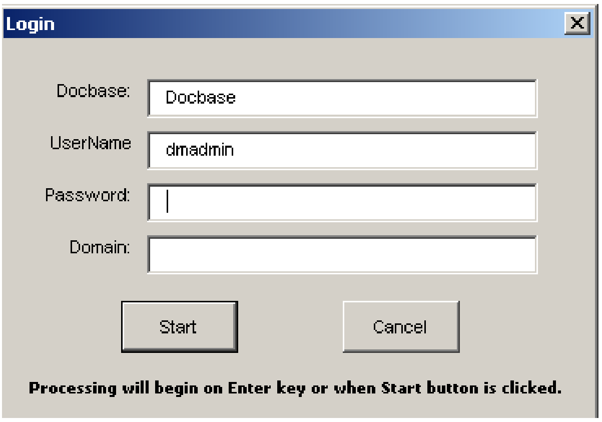
A	B	C	D	E	F	G
ToolSS Install Path	C:\ToolSS\Create Groups	**Required**	Full path specification where CreateGroups.jar is deployed.		**Group Class**	group
CLASSPATH	C:\ToolSS\Create Groups\CreateGroups.jar;%CLASSPATH%	**Required**	Default uses ToolSS Install Path and system configuration.			role
Temp Path	C:\Temp Test	**Required**	Path where generated input and output files will be saved.			module role
Docbase Name	DCTM65	**Optional**	Default for login dialog. Case sensitive.			privilege group
User Name	dmadmin	**Optional**	Default for login dialog. Case sensitive			domain

Click Create button for login dialog.
Clicking Start button on subsequent login dialog begins processing.

Enter the required configuration information.

Tip: The Group Class column (G) provides a list of valid group_class values as defined for Documentum 6.5. The first line of the Input worksheet contains a data validation control (drop down list) that references this column. It can be copied to other rows as desired.

When ready to being processing, click the Create button. This will present a login dialog:



Docbase: Docbase
UserName dmadmin
Password:
Domain:

Processing will begin on Enter key or when Start button is clicked.

The Docbase and UserName fields are populated by default based on the entries in the Config&Run worksheet (if entered); otherwise they must be entered manually. Enter password. Use the Enter key or Start button to begin processing.

Note: no further warnings are provided once Enter key or Start button are activated.

When processing is started a command window will display during CreateGroups execution. When processing is complete, a summary of the execution will be presented, such as the following:

```
CreateGroups Started 03-14-2010 17:00:17
Session started in repository DCTM65for user dmadmin.

Processing record 1: Root Group,Group 1,Group 2,Group 3,Group 4,Group 5,Group 6,Group
7,Group 8,Group 9,Group 10,Description,Owner Name,Group Admin,Email Address,Group Class
Processing record 2: hr_all,,,,,,,,,,,All users in Human Resources
Department,dmadmin,hr_admin,dummy,group
Processing record 3: ,hr_read,,,,,,,,,,Users with read access to Human Resources
Documents,dmadmin,hr_admin,dummy,group
Processing record 4: ,hr_write,,,,,,,,,,Users with write access to Human Resources
Documents,dmadmin,hr_admin,dummy,group
Processing record 5: ,hr_admin,,,,,,,,,,Users with full administrative access to Human
Resources Documents,dmadmin,admingroup,dummy,group
Processing record 6: acct_all,,,,,,,,,,,All users in Accounting
Department,dmadmin,acct_admin,dummy,group
Processing record 7: ,acct_read,,,,,,,,,,Users with read access to Accounting
Documents,dmadmin,acct_admin,dummy,group
Processing record 8: ,acct_write,,,,,,,,,,Users with write access to Accounting
Documents,dmadmin,acct_admin,dummy,group
Processing record 9: ,acct_admin,,,,,,,,,,Users with full administrative access to
Accounting Documents,dmadmin,admingroup,dummy,group

CreateGroups Finished 03-14-2010 17:00:43
Records Processed: 9
Records Successful: 9
Records with Errors: 0
Records with Warnings: 0
See report for details: C:\Temp Test\CreateGroups-2010-03-14-17-00-17.log
```

At completion, three files will be found in the defined Temp Path:

CreateGroups-<yyyy-mm-dd-hh-mi-ss>.csv [Generated Input File]
CreateGroups-<yyyy-mm-dd-hh-mi-ss>.out [Execution Summary]
CreateGroups-<yyyy-mm-dd-hh-mi-ss>.log [CreateGroups Execution Log]

The CreateGroups Execution Log is described in the next section.

Command Line Mode

Command Line mode allows for preparation of the input file separately from execution of the CreateGroups class. For example, the input file can be prepared from any workstation, and then used as input for running on another workstation where DFC and CreateGroups are deployed. Thus, this is a two step process: File Preparation and Command Line Execution.

File Preparation

Once the spreadsheet is configured as desired and reviewed and approved, it must be converted to CSV format for use with the CreateGroups utility via Command Line Mode. If no modifications to the layout have been made and no additional data exists to the right or below input data, then the file can be converted via the File > Save As > CSV menu option within Excel.

Alternatively, the following is the recommended approach for preparing CSV files for the CreateGroups class. This approach helps to avoid input data processing issues. This will also avoid having to save the original spreadsheet as CSV (which can lead to confusion if modified after saving as CSV).

1. Select upper left-most cell (A1).
2. <shift> select lower right-most cell with data (e.g. the Object Type column for the last row of data). Desired data should now be highlighted.
3. Edit > Copy.
4. Open new worksheet.
5. Select upper left-most cell (A1).
6. Edit > Paste.
7. File > Save As > CSV. (ignore warnings about format and force the save).
8. Close the new worksheet (ignore any warnings about saving changes at this point).
9. Copy CSV to desired location for execution of CreateGroups utility.

Tip: After the input file is run through the CreateGroups class, rename it using the same name as generated for the utility report (with timestamp). Or, vice-versa, rename the report to the same name as the input file (but retaining the .log extension). With either approach, it makes it easier to tell which input file went with which report.

Command Line Execution

CreateGroups is a command line utility. Parameters are passed as space-separated arguments using the following generic syntax for the class:

```
CreateGroups [docbase] [username] [password] [input file spec]
```

The parameters must be passed in the order specified and described below:

Order	Name	Description
1	docbase	Name of the repository. Note: Case sensitive
2	username	Username for repository session.
3	password	User's password for repository session. Note: not masked.
4	input file spec	The full file path and name specification for the input file. Note: if the path or name contains spaces, this must be enclosed in double quotes.

Example generic syntax for connection to TEST repository as MyUser for input file named myinput.csv in C:\temp:

```
CreateGroups TEST MyUser ******** C:\temp\myinput.csv
```

In order for the CreateGroups class to function properly, it may be necessary to add classpath information to the command line syntax. This depends on your user and machine configuration. Minimally, the tool must be able to resolve the classpath to:

- Standard Java classes in installed JRE or JDK (typically, these would be available via your system or user CLASSPATH configuration)
- dfc.jar (typically configured under DCTM shared directory)
- CreateGroups.jar (directory where this utility was deployed to)

The following provides an example of the full command line syntax (Windows) when running from the directory where the CreateGroups.jar file was deployed to, assuming JRE and DCTM resources are available via CLASSPATH environment variable:

```
java -classpath .;.\CreateGroups.jar;%CLASSPATH% com.sssc.toolss.cf.CreateGroups TEST MyUser
***** C:\temp\myinput.csv
```

The following shows a similar example for UNIX:

```
java -classpath .:./CreateGroups.jar:$CLASSPATH com.sssc.toolss.cf.CreateGroups TEST MyUser
******** /temp/myinput.csv
```

Note: if a java.lang.NoClassDefFoundError is returned when attempting to run the utility, it is most likely a classpath configuration issue.

Note: if a parameter contains a space character, it must be enclosed within double quotes.

Once the utility begins execution, information will be displayed via standard output (e.g. command console). A line is printed for each input record processed, so you can track execution progress. At the conclusion of execution, a summary of information is displayed similar to the following:

```
CreateGroups Finished 03-14-2010 17:00:43
Records Processed: 9
Records Successful: 9
Records with Errors: 0
Records with Warnings: 0
```

Additionally, an execution log is generated in the directory from where the utility was initiated at the command prompt. It will be named using the following convention:

```
CreateGroups-MM-DD-YYYY-HH-mm-ss.log
```

This naming convention will ensure a unique filename for each execution. This log will contain:

1. Header section with the class name, start time, and input info.
2. Record section with metadata for all input records processed.

3. Summary section with stop time and summary counts (processed successful, errors, warnings).

The Record section contains the following information in CSV format (this allows opening of the reports in MS Excel [by changing the extension to .csv] for easy review and manipulation in spreadsheet format):

"Record Count","Status","Group Path","Message"

Record Count	Running count of input records processed.
Status	INFO: Used for header row only (skipped). SUCCESS: group created successfully.. ERROR: Failed to create group due to exception.
Group Path	The full group path for the input record. N/A for header record.
Message	If Status = SUCCESS, then the r_object_id of the group. If Status = INFO, then "Skipping header record." If Status = ERROR, then Exception message.

Other Processing Notes

- The CreateGroups utility only creates groups, as its name implies. It will never delete groups or remove assignments. Thus, the utility does not perform any checking or cleanup to ensure that any extra groups that are not specified in the input file do not exist. For example, if a group named Test already exists with three subgroups named Test1, Test2, and Test3, and the input file has group structure defined for Test group with three subgroups Test4, Test5, and Test6; then, at the end of execution the Test group will have all six subgroups. The CreateGroups utility would only add the groups defined in the input file because they do not already exist or have not been assigned. It does not check or attempt to remove the previously existing groups Test1, Test2, and Test3 even though they are not specified within the input file. Therefore, it is recommended to review what exists within the repository before running the utility. It some cases, it may be desirable to move any existing groups from similar structure defined within the input file (such as Test1, Test2, Test3 groups) to ensure that you wind up with only the group structure defined within the input file.
- If a group already exists, its creation will be skipped. If a group assignment already exists, it will also be skipped. Thus, this allows for the use of CreateGroups for the creation of nested group structure for groups that may already exists. Additionally, a group can be referenced more than once in multiple group structures if desired.

Known Issues

Single Quotes

During testing, it was discovered that the utility does not support the use of single quote characters within group names, although this is permitted within a repository. Typically the use of single quote characters within names are discouraged, as this can lead to query and processing errors. If single quotes are used, the following exception will be encountered:

```
Problem processing group record: [DM_API_E_QUERY_FAIL]error:  Query failed:
select r_object_id from dm_group where any r_group_path = '<your path with single
quote>'
```

Support for single quotes may be added as a future enhancement. Until then, it is necessary to remove any single quotes.