Paper SAS1779-2015

Row-Level Security and SAS® Visual Analytics

Zuzu Williams, SAS Institute Inc., Cary NC

ABSTRACT

How can you protect your data used by SAS® Visual Analytics considering the challenges of data security in today's business environment? SAS has implemented security features in its widely used business intelligence platform, including row-level security in SAS Visual Analytics. Row-level security specifies who can access particular rows in a SAS LASR Analytic Server (LASR) table. Throughout this paper, we discuss two ways—interactively and in batch—of implementing row-level security for LASR tables in SAS Visual Analytics. Both approaches link table-based permission conditions with metadata stored identities.

INTRODUCTION

Row-level security in SAS Visual Analytics refers to selectively granting the ability to use particular rows within the LASR tables used in SAS Visual Analytics explorations and reports. To end users who have restricted access to certain rows, it is as if only those rows exist. These permission conditions are stored in the SAS® Metadata Server.

This paper explains the basics of administering row-level security both from the SAS Visual Analytics user interface as well as from batch tools. And we offer suggestions on how to verify that permissions conditions are truly in effect. Furthermore, you will learn under what circumstances the row-level security you have applied will be maintained, even after LASR tables have been removed from the memory.

Examples in this paper are created in SAS Visual Analytics 7.1 in a non-distributed environment on a Windows server. The examples use the SASHELP.PRDSALE data set that is supplied with SAS® 9.

PREREQUISITES

It is important to note that permission conditions must be applied on a per LASR table basis. Also, each LASR table must be registered in the SAS Metadata Server. For star schemas that are used in LASR, the permission condition must be applied to the entire schema, not to the individual tables within it.

Users for whom you want to apply permission conditions must have a SAS metadata identity—that is, they must exist in the SAS metadata environment. Permissions on a LASR table can be set at group level or user level. For administration and ease of maintenance, group identities must be created in the SAS Metadata Server.

To access and use SAS Visual Analytics, users must belong to a SAS Visual Analytics group (or groups). To set row-level security on a LASR table, an administrator must belong to a SAS Visual Analytics Data Administrators group. Users must belong to a SAS Visual Analytics Users group or a Visual Data Builder Administrators group (or both) for data building.

GETTING STARTED

Row-level security is set by adding constraints called permission conditions to the explicit grants of the Read permission for the LASR table. Permission conditions are defined by data filter expressions.

Each permission condition filters a particular LASR table based on what content each user or group should or should not see.

There are three possible outcomes:

- **Grant** – The user can see all rows.
- **Conditional grant** – The user can see rows that meet specified filtering conditions.
- **Denial** – The user can't see any rows.

In the event of conflicting grants, a conditional grant on a parent object (group) takes precedence over a user's explicit grants.

To apply row-level security the LASR table doesn't have to be in LASR memory, it has to be registered in the SAS Metadata Server.

APPLYING ROW-LEVEL PERMISSONS INTERACTIVELY

Row-level security can be set interactively using either of the following two methods:

- You can use point-and-click functionality on the **Visual** tab.
- You can use the enhanced editor on the **Text** tab.

Visual Tab

1. First, on the SAS Visual Analytics Home page, click the Administrator icon.

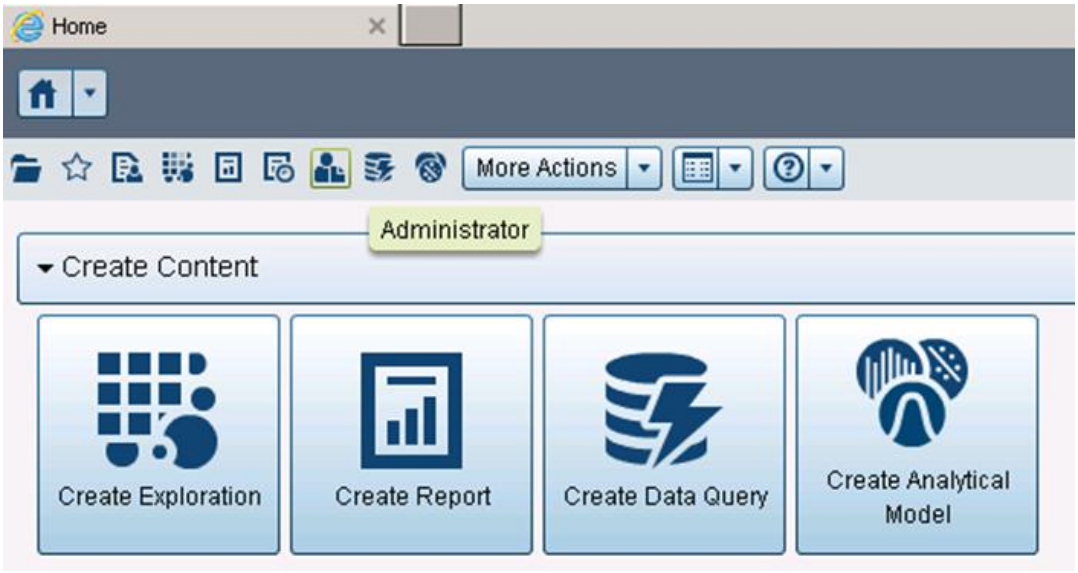

Figure 1: Select the Administrator Icon

2. Next, on the Manage Environment page, right-click the LASR table to get to the Authorization page. Be sure to navigate to the LASR folder location to select the LASR table and not to the INPUT table.

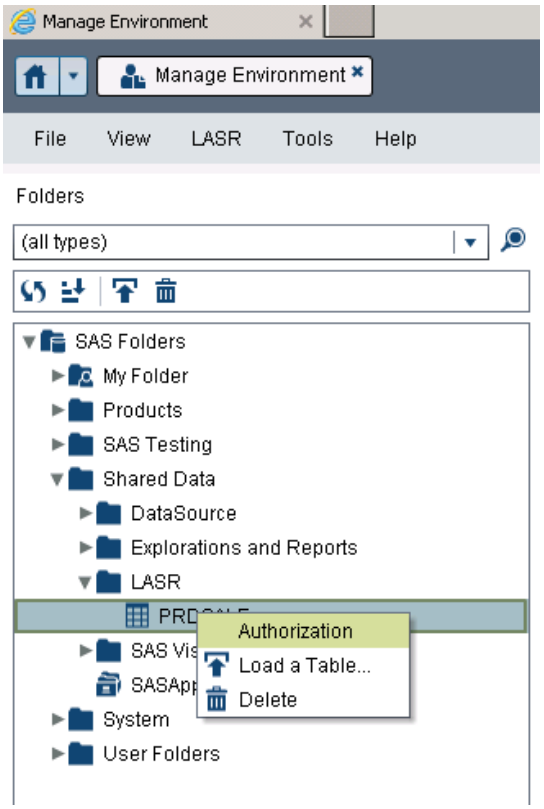

Figure 2. Authorization on a Table

3. On the Authorization page, you have to add identities first. You can select multiple users or groups at once. Then modify the user Read permissions to include Conditional grant.

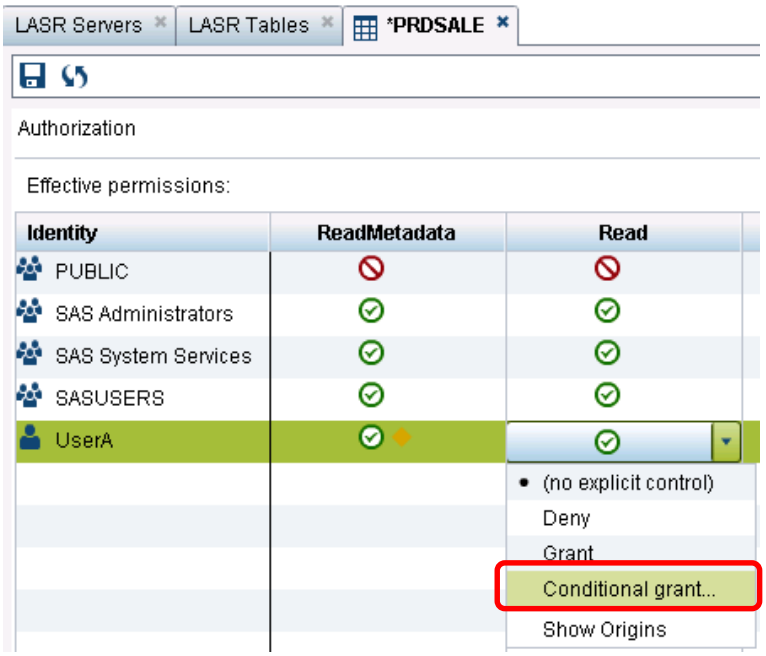

Figure 3. Modify Read Permission

4. In the New Permission Condition window, select the Visual tab to specify your condition.

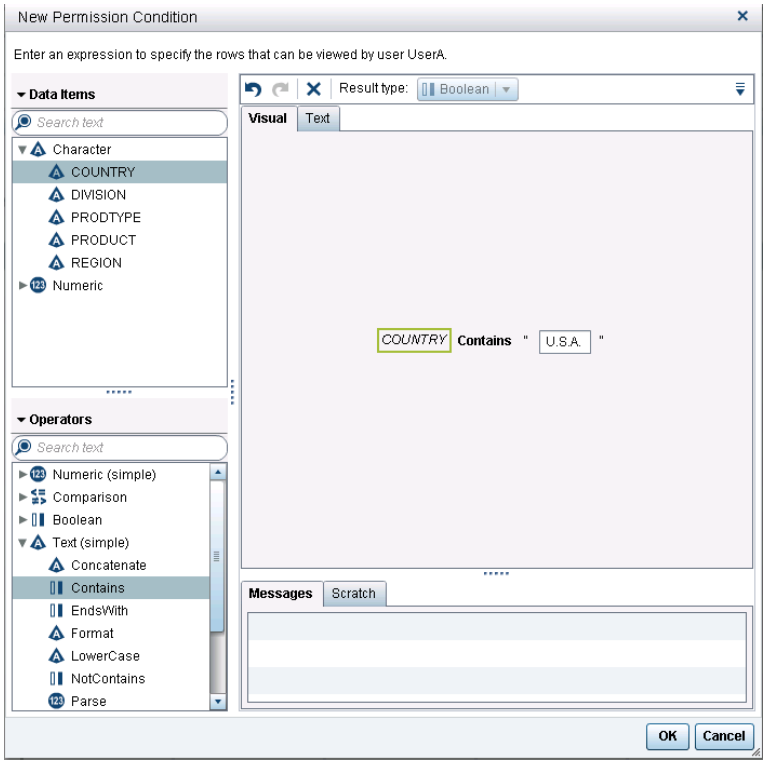

Figure 4. Edit Permission Condition Window

5. It is important to save the changes you made to the object, otherwise your changes won't take effect. In this example, UserA's Read permission is modified with a conditional grant to include only rows with Country values of "U.S.A.".

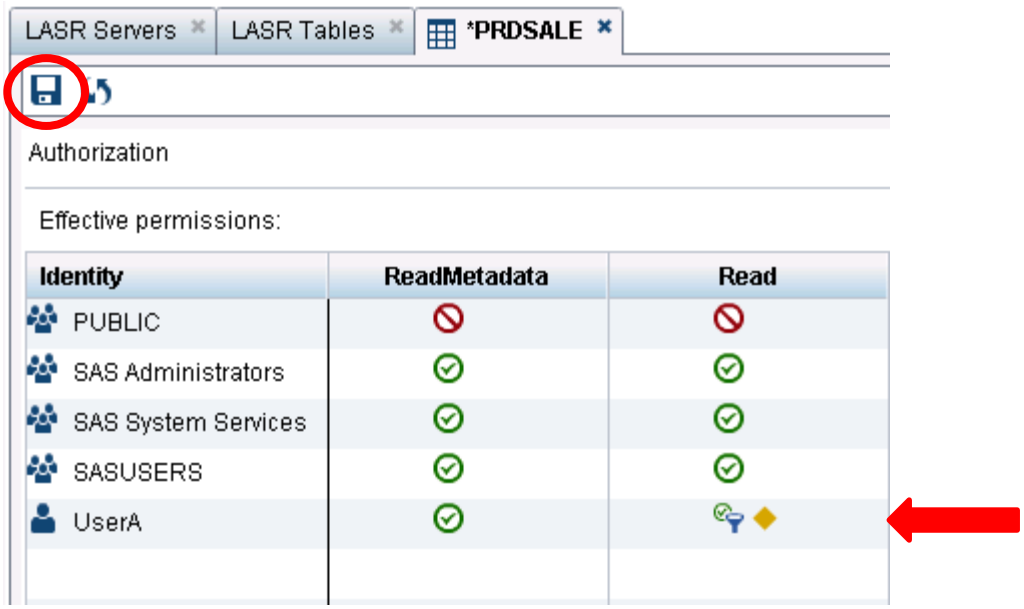

Figure 5. Conditional Grant Is applied for UserA on the PRDSALE Table

Text Tab

1. To use the **Text** tab option, follow steps 1 through 3 just as in the previous section.
2. In the New Permission Condition window, click the **Text** tab and enter your expression.

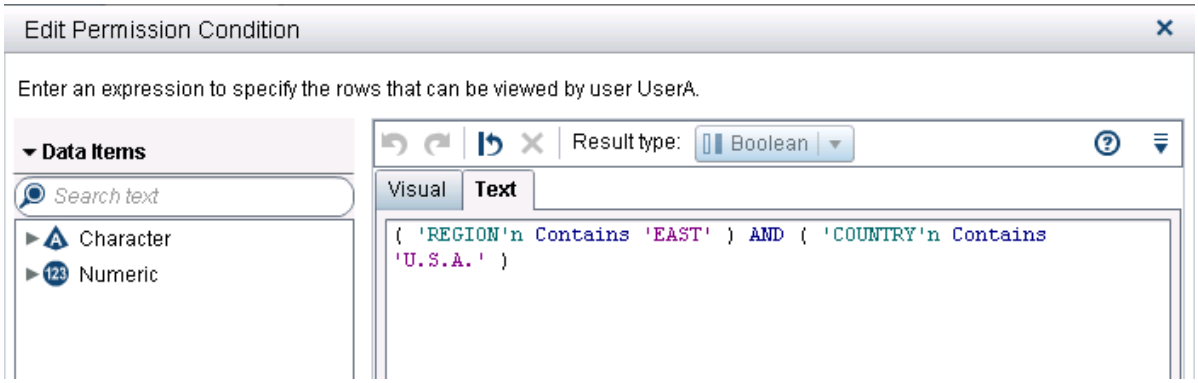

Figure 6. Conditional Grant Using an Expression in the Text Tab.

Now, UserA can see sales numbers for all product types only in East region. USA.

Country	Region	Division	Product	Predicted Sales	Actual Sales	Quarter	Year
U.S.A.	EAST	CONSUMER	BED	$10,941.00	$13,211.00	3	1994
			CHAIR	$11,958.00	$15,485.00	3	1994
			DESK	$14,349.00	$11,068.00	3	1994
			SOFA	$12,471.00	$9,707.00	3	1994
			TABLE	$11,257.00	$11,803.00	3	1994
		EDUCATION	BED	$12,657.00	$9,926.00	3	1994
			CHAIR	$11,394.00	$11,893.00	3	1994
			DESK	$12,190.00	$12,125.00	3	1994
			SOFA	$9,963.00	$12,556.00	3	1994
			TABLE	$13,407.00	$10,455.00	3	1994

Figure 7. View in VA Explorer with Row-Level Security Applied

You can make the user's permissions even more granular by specifying additional syntax such as the following:

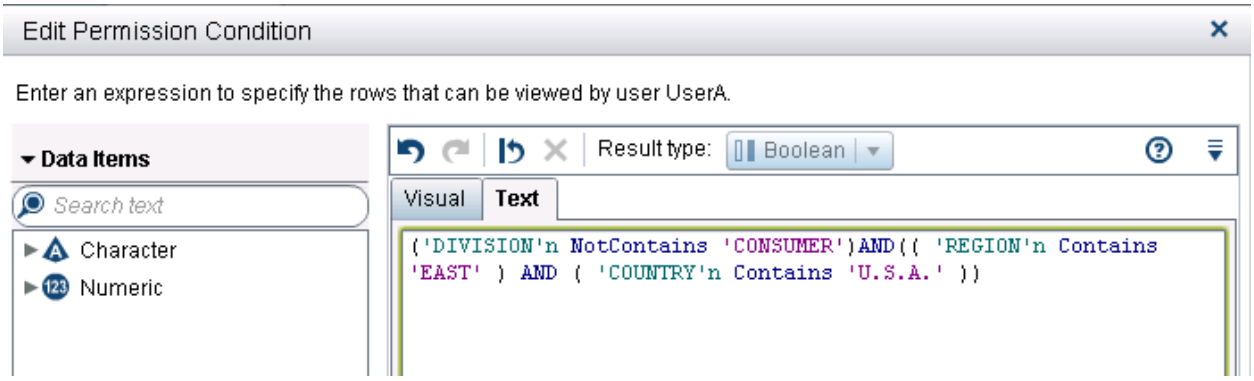

Figure 8. Edit Permission Condition

UserA can now see sales numbers only for the Education division.

Country	Region	Division	Product	Predicted Sales	Actual Sales	Quarter	Year
U.S.A.	EAST	EDUCATION	BED	$12,657.00	$9,926.00	3	1994
			CHAIR	$11,394.00	$11,893.00	3	1994
			DESK	$12,190.00	$12,125.00	3	1994
			SOFA	$9,963.00	$12,556.00	3	1994
			TABLE	$13,407.00	$10,455.00	3	1994

Figure 9. View with Edited Permissions

APPLYING ROW-LEVEL PERMISSIONS IN BATCH USING THE sas-set-metadata-access TOOL

Now let's suppose that not only do you need to specify conditions based on country and region, but also based on various departments in your organization. What if you want to give various employees with the same department different access to the same data resource based on their job roles? How can you achieve this without spending too much time defining security for each and every one of them?

Luckily, SAS provides batch utilities for metadata-layer access control to help you with this. Batch tools are installed with every SAS Visual Analytics implementation.

The batch tools for access control are Java scripts that enable you to connect to a metadata server and perform the following tasks:

- View, set, or remove direct access controls (explicit grants, explicit denials, permission conditions, and directly applied access control templates) for a specified object.
- Create a new access control template, update an existing access control template, and display an ACT's permission pattern.

Batch utilities for metadata-layer access control are located in the following path:

`SAS-installation-directory/SASPlatformObjectFramework/9.4/tools`

Syntax

General form of the command below:

sas-set-metadata-access *connection-options object-path special-options*

Option	Description
-host host-name	Identifies the host of the metadata server. This option is required if the -profile option is not set.
-password password	Specifies the password of the connecting user. This option is required if the -profile option is not set.
-port port	Specifies the port on which the metadata server runs. This option is required if the -profile option is not set.
-profile profile-name	Specifies a connection profile. You can use this option instead of using individual options (-host, -port, -user, and -password) to provide connection information.
-user user-ID	Specifies the user ID of the connecting user. This option is required if the -profile option is not set.

Table 1. Connection Options for Metadata Batch Tools

Option	Description
-condition *condition-expression*	Adds the specified expression (as a permission condition) to the immediately preceding explicit grant.
-deny *identity-name*:*permission1*<,*permission2*...>	For the specified user or group, sets explicit denials of one or more permissions.
-grant *identity-name*:*permission1*<,*permission2*...>	For the specified user or group, sets explicit grants of one or more permissions.
-remove *identity-name*:ALL	For the specified user or group, removes all explicit grants or denials.
-remove *identity-name*:*permission1*<,*permission2*...>	For the specified user or group, removes explicit grants or denials of one or more permissions.
-removeAll –removeAll	Removes all direct access controls (explicit grants, explicit denials, permission conditions, and directly applied ACTs).

Table 2. Special Options for sas-set-metadata-access

Batch tools must run on a machine from which the metadata server associated with an instance of SAS Visual Analytics can be accessed.

Examples

Grant access to UserA for countries: USA and Canada:

```
sas-set-metadata-access -profile Admin "/Shared Data/LASR/PRDSALETable)" -
grant userA:Read -condition "(Country='CANADA') AND (Country='U.S.A.')"
```

Grant access to users in GroupA for countries USA and Canada:

```
sas-set-metadata-access -profile Admin "/Shared Data/LASR/PRDSALE(Table)" -
grant groupA:Read -condition "Country IN ('U.S.A.', 'CANADA')"
```

Grant access to users in GroupB for regions East and West:

```
sas-set-metadata-access -profile Admin "/Shared Data/LASR/PRDSALE(Table)" -
grant groupB:Read -condition "Region IN('EAST','WEST')"
```

You can add as many conditions as you would like to your batch file and then run it based on a specific schedule or ad hoc.

VERIFYING ROW-LEVEL PERMISSIONS USING THE sas-show-metadata-access TOOL

You can use sas-show-metadata-access tool to view access permissions as well as conditional grants. Just as the sas-set-metadata-access tool, this tool is also part of the batch tools for metadata authorization utility. It can be executed in batch or on the command line.

Syntax

General form of the command below:

sas-show-metadata-access *connection-options object-path <special-options>*

For specifics on connection options please refer to Table2.

Option	Description
-effective	Displays effective access
-onlyGroup group-name	Displays access for specific user group
-onlyUser user-name	Displays access for specific user

Table 3. Special Options for sas-show-metadata-access

Examples

For the **/Shared Data/LASR/PRDSALE** object, show all direct access controls that are assigned to the GroupA:

```
sas-show-metadata-access -profile SASAdmin "/Shared Data/LASR/PRDSALE(Table)"
-onlyGroup groupA
```

Here is some example output from the preceding command:

```
  -grant "GroupA(UserGroup)":Read
    -condition "Country IN ('U.S.A.', 'CANADA')"
```

Show all access controls associated with the **/Shared Data/LASR/PRDSALE** object:

```
sas-show-metadata-access -profile SASAdmin "/Shared Data/LASR/PRDSALE(Table)"
```

Here is the example output:

```
-grant "GroupA(UserGroup)":Read
```

```
    -condition "Country IN ('U.S.A.', 'CANADA')"
  -grant UserA:Read
    -condition "Country IN ('U.S.A.')"
```

VERIFYING ROW-LEVEL PERMISSIONS USING PROC IMSTAT

To test row-level security programmatically, you can use nonstatistical functionality available in PROC IMSTAT. This procedure is accessible with Base SAS®, which is included with SAS Visual Analytics.

Concept

Use the syntax available in PROC IMSTAT, which operates on LASR tables, to determine the number of rows that a user can or cannot see. This is achieved by constructing a filtering WHERE clause that mimics effective row-level permission conditions, and is reported via the NUMROWS statement that accompanies it.

The beauty of this test is that all of the work for this query is done in LASR memory. No data is moved out of LASR, and there is no need to create new or temporary tables within LASR.

> **CAUTION: If you are considering using PROC SQL or other pre-LASR SAS tools to do your testing, be aware you might be moving large amounts of data out of LASR.** This approach is generally not efficient because it raises performance and resource-use concerns if your LASR tables are large.

Example

```
/* Submit libname statement to the SAS LASR Analytic Server */
LIBNAME VALIBLA SASIOLA  TAG=HPS  PORT=10010 HOST="server.example.com"
SIGNER="http://server.example.com:80/SASLASRAuthorization" ;

/* Establish connection to the SAS Metadata Server as the test user */
options metauser="TestUser"  metapass= "XXXXXX"  metaserver="server.example.com"
metarepository="Foundation";


/* Using proc imstat, query LASR table for those rows that test user does not have
permission to see.
 * If table is secured as expected, then NUMROWS will report a value of ZERO for
denied rows */
proc imstat;
       table valibla.prdsale;   /* table with row level security applied */

              /* In this example, our test user has been denied access to all rows
except those
               * where country = " U.S.A."
               */
              where country ne "U.S.A.";

              /* NUMROWS should return zero */
              numrows;
run;

/* Next, the same table is queried for non-denied rows; we expect a value of > zero,
if there are rows
 * where country equals "U.S.A.". The valibla.prdsale table remains the active table
between "run" statements.
 */
where country eq "U.S.A.";
```

```
/* NUMROWS returns number of rows that user has permission to view, which is expected
to > 0 */
numrows;
run;

quit;
```

SAS Output

Results from query where country ne “U.S.A.”

The SAS System

The IMSTAT Procedure

Number of Rows Action for Table HPS.PRDSALE	
	Number of Records
	0

Results from query where country eq “U.S.A.”

The SAS System

The IMSTAT Procedure

Number of Rows Action for Table HPS.PRDSALE	
	Number of Records
	480

ROW-LEVEL SECURITY - LIFETIME

How often do you need to apply row-level security and how long does it last?

Let’s consider the following scenario:

Your INPUT table is loaded into LASR memory, and row-level permission conditions are applied. Next, your LASR table gets unloaded from LASR memory. What then happens to the row-level security?

> As long as your LASR table is registered in the SAS Metadata Server, you can simply re-load the table back into LASR memory and retain the row-level permissions. If for some reason metadata representation of your LASR table is deleted, then permission conditions will be lost. You have to reload the INPUT table into LASR memory and re-apply the row-level security conditions again.

CONCLUSION

This paper discussed two methods to apply row-level security for the LASR tables. The interactive method is simple enough as long as you only need to set permissions on tables for small groups of employees within your organization. However, this approach is not efficient if you have a large enterprise with many different permission conditions on many tables. It is in this situation that batch tools can be very useful. Choose the method that works best for you. In either case, row-level permissions will be in effect for the lifetime of your LASR tables as long as they do not get deleted from the SAS Metadata Server.

RECOMMENDED READING

- *SAS® 9.4 Intelligence Platform: Security Administration Guide.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation/cdl/en/bisecag/67045/PDF/default/bisecag.pdf.
- *SAS® LASR™ Analytic Server 2.4: Reference Guide*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation/cdl/en/inmsref/67597/PDF/default/inmsref.pdf.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Zuzu Williams
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Zuzu.Williams@sas.com
www.sas.com