



Introduction to Custom Processing for SAS[®] Marketing Automation

Course Notes

Introduction to Custom Processing for SAS® Marketing Automation Course Notes was developed by Lise Cragen and Donna LeBlanc. Additional contributions were made by Steve Marshall. Editing and production support was provided by the Curriculum Development and Support Department.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

Introduction to Custom Processing for SAS® Marketing Automation Course Notes

Copyright © 2017 SAS Institute Inc. Cary, NC, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Book code E71207, course code ATECIMA, prepared date 11Dec2017.

ATECIMA_001

Table of Contents

To learn more.....	iv
Chapter 1 Introduction to Custom Processing for SAS® Marketing Automation.....	1-1
1.1 Introduction to Custom Processing	1-3
Demonstration: Registering a Stored Process Using SAS Enterprise Guide.....	1-23
1.2 Learn More about Custom Processing.....	1-26

To learn more...



For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to training@sas.com. You can also find this information on the web at <http://support.sas.com/training/> as well as in the Training Course Catalog.

For a list of SAS books (including e-books) that relate to the topics covered in this course notes, visit <https://www.sas.com/sas/books.html> or call 1-800-727-0025. US customers receive free shipping to US addresses.

Chapter 1 Introduction to Custom Processing for SAS[®] Marketing Automation


1.1 Introduction to Custom Processing.....	1-3
Demonstration: Registering a Stored Process Using SAS Enterprise Guide	1-23
1.2 Learn More about Custom Processing.....	1-26

1.1 Introduction to Custom Processing

What We'll Cover Today

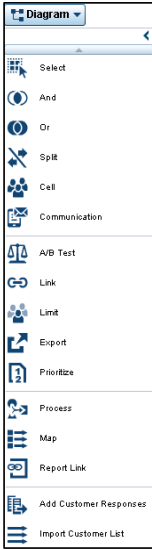
Nodes that perform custom processing	Steps to create a stored process
Processing resources	Example applications
Coding best practices	Demonstration of stored process registration

6




Copyright © SAS Institute Inc. All rights reserved.

Custom SAS Processing for Marketing Automation



The screenshot shows a vertical menu of process nodes in SAS. The nodes listed are: Select, And, Or, Split, Cell, Communication, A/B Test, Link, Limit, Export, Prioritize, Process, Map, Report Link, Add Customer Responses, and Import Customer List. Each node is accompanied by a small icon representing its function.

8




Copyright © SAS Institute Inc. All rights reserved.

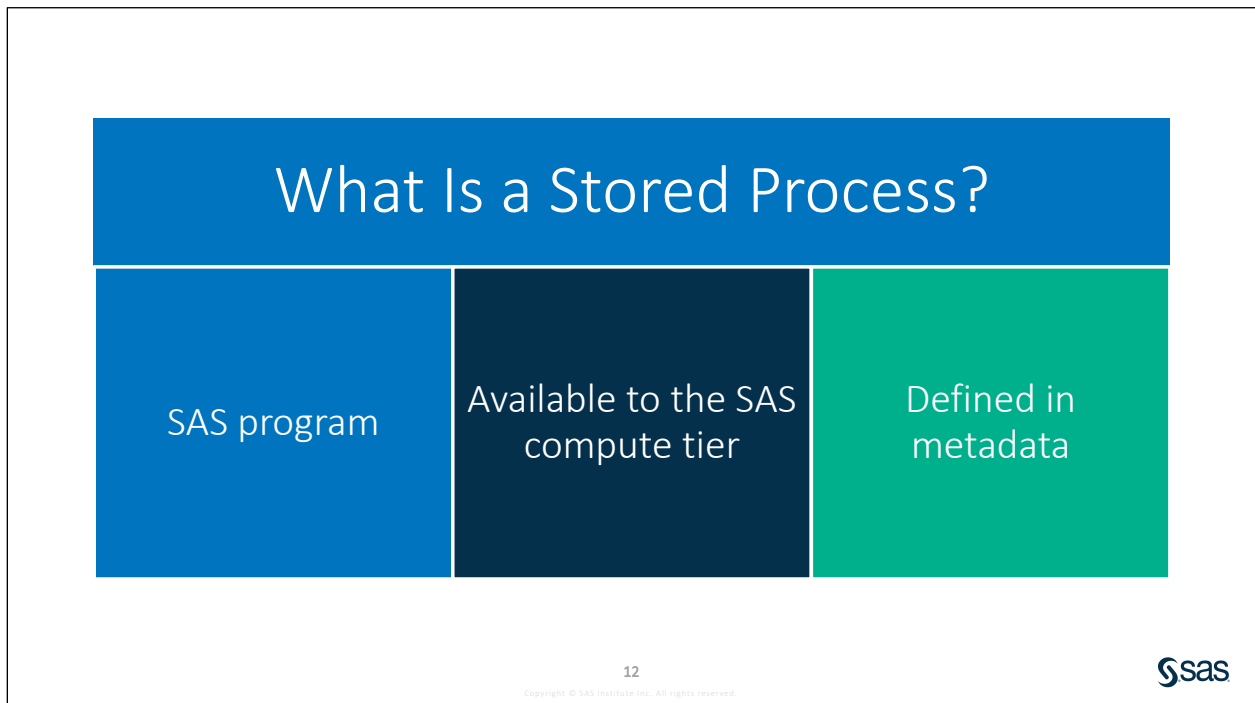
In a Marketing Automation campaign, you can execute a SAS program that has been written to perform a specific task related to campaign execution. You can use two different nodes for this purpose: the Process node or a Custom node.

	Process Node	Custom Node
SAS Stored Process	✓ Best Practice	✓
SAS Code	✓	✗ Not Available

11
Copyright © SAS Institute Inc. All rights reserved.



This presentation focuses on SAS Stored Processes. Using a SAS Stored Process is a best practice for the Process node and the only option available for a Custom node.



Writing Stored Processes: Knowledge Needed

Need to know:

- SAS Language
- Macro language

Good to know:

- SQL

13

Copyright © SAS Institute Inc. All rights reserved.



Creating a Stored Process for Marketing Automation

Step 1



Write and test the stored process code.

You should test your code outside of SAS Customer Intelligence Studio.

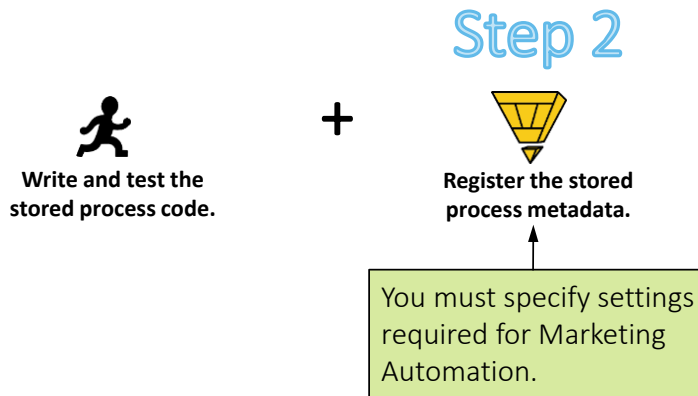
You must include macro calls required for Marketing Automation.

14

Copyright © SAS Institute Inc. All rights reserved.



Creating a Stored Process for Marketing Automation



15

Copyright © SAS Institute Inc. All rights reserved.



Creating a Stored Process for Marketing Automation



16

Copyright © SAS Institute Inc. All rights reserved.



Write the Code

Your stored process code can include macro variable references and macro program calls that are specific to Marketing Automation.

&MacroVariable

%MacroCall

17

Copyright © SAS Institute Inc. All rights reserved.



Write the Code

A stored process program includes the original SAS code as well as additional SAS statements and macro calls.

SAS Program

```
proc sql noprint;
select name, description into
:nodename, :nodedesc
from work.inputnodes
...
```

Stored Process Program

```
%stpbegin
%maspinit(xmlstream= macrovar neighbor)

proc sql noprint;
select name, description into
:nodename, :nodedesc
from work.inputnodes
...

%macnttab
%mastatus( &_stpwork.status.txt )

%stpend
```

18

Copyright © SAS Institute Inc. All rights reserved.



%Stpbegin and **%Stpend** are required for all stored processes, including those written for use with Marketing Automation.

The call to **%Mastatus** shown above is required for all Marketing Automation stored processes. It must appear immediately before the call to **%Stpend**.

Calls to **%Maspinit** and **%Macnttab** (or **%Macount**) are required in certain circumstances for Marketing Automation stored processes.

Register the Metadata

Required:

- MAUser keyword (case sensitive)
- Result Capabilities = Package

Might be needed:

- macrovar and Neighbor data streams
- Prompts
- Keywords to specify subjects
- Keywords to control number of output cells

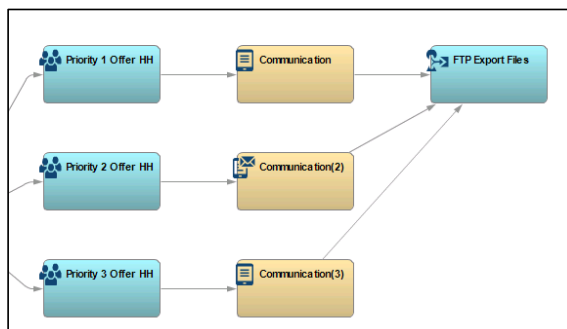
19

Copyright © SAS Institute Inc. All rights reserved.



Example 1: FTP Export Files

A stored process moves export files created by communication nodes to another server for further processing.



20

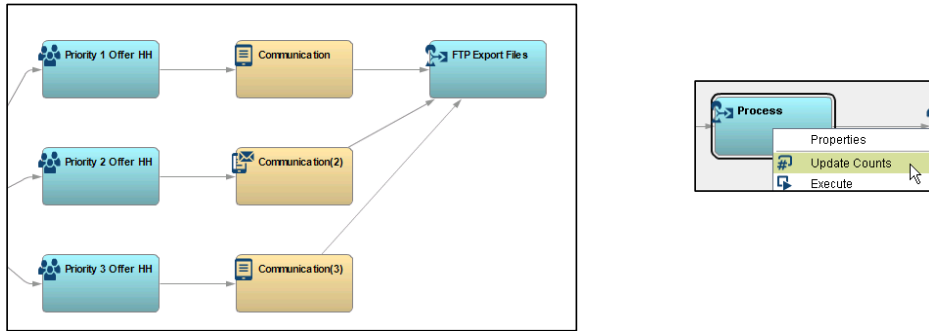
Copyright © SAS Institute Inc. All rights reserved.



Example 1: FTP Export Files

Requirement 1:

The stored process should move the export files only when the campaign is executed, not when counts are updated.

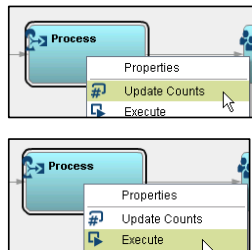


21

Copyright © SAS Institute Inc. All rights reserved.



Macro Variable: COUNTONLY



→ &COUNTONLY = Y

→ &COUNTONLY = N

```
%if &countonly=N %then %do;
    /* Move the files */
%end;
```

22

Copyright © SAS Institute Inc. All rights reserved.

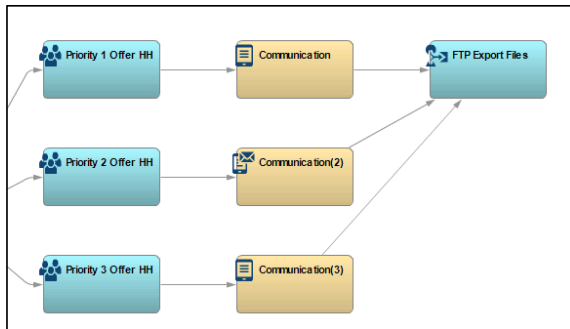


Beginning in Marketing Automation 6.1, you can use the **&COUNTONLY** macro variable to control the behavior of the Process or Custom node depending on whether **Update Counts** or **Execute** was selected. You can design your stored process to update counts only and not perform its complete operation when the value is Y.

Example 1: FTP Export Files

Requirement 2:

This stored process is designed to access information from Communication nodes upstream. If there are none, we want campaign execution to fail with a custom error message for troubleshooting purposes.



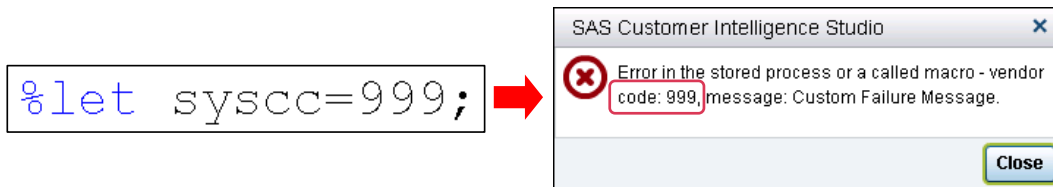
23

Copyright © SAS Institute Inc. All rights reserved.



Macro Variable: SYSCC

- You can set the value of **SYSCC** to any value greater than 4 to cause the campaign to fail.
- Failure messages, including the value of **SYSCC**, are written to the Customer Intelligence core log.
- For campaigns that are manually executed in Customer Intelligence Studio, an error dialog box showing the value of **SYSCC** appears.



24

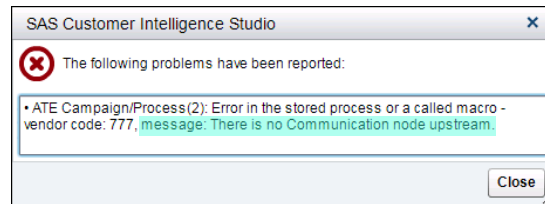
Copyright © SAS Institute Inc. All rights reserved.



Macro Variable: MAMSG

- **MAMSG** is a macro variable that is specific to SAS Marketing Automation.
- It enables you to specify a custom message to be displayed in the Marketing Automation core log and Customer Intelligence Studio error dialog box.

```
%let mamsmsg=There is no Communication node upstream;
```



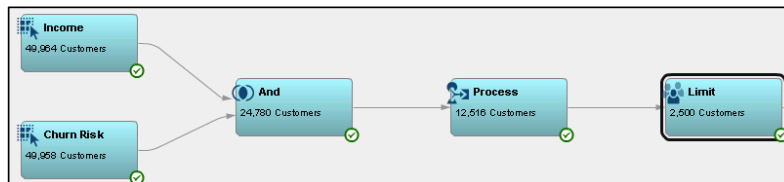
25

Copyright © SAS Institute Inc. All rights reserved.



Example 2: Apply a Propensity Model

- The Process node applies a propensity model to subjects meeting the criteria of the And node.
- The scored subjects are passed to the Limit node.



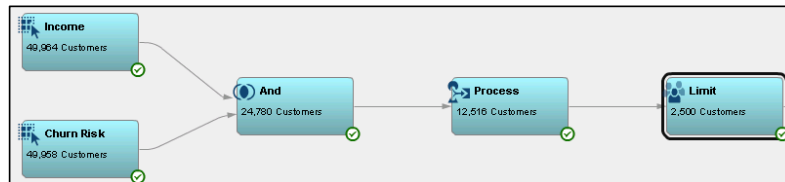
26

Copyright © SAS Institute Inc. All rights reserved.



Example 2: Apply a Propensity Model

- Requirements:
 - Identify subjects selected by upstream nodes.
 - Pass results on to downstream nodes.



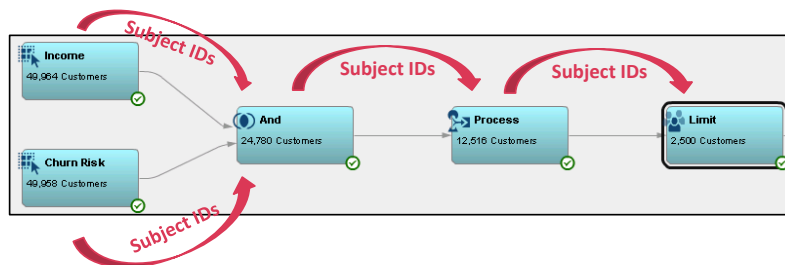
27

Copyright © SAS Institute Inc. All rights reserved.



MATables Library

- To expedite processing, SAS Marketing Automation automatically creates tables behind the scenes containing the IDs of qualified subjects.
- These are stored in a library named **MATables**.
- You can read from and write to this library.



28

Copyright © SAS Institute Inc. All rights reserved.



MATables is a SAS library that contains SAS data sets created by SAS Marketing Automation. Data sets in this library are automatically created on the SAS Server as an intermediate query result of qualified subjects. These data sets have the following characteristics:

- names are generated by Marketing Automation
- contain a list of subjects
- include the subject ID and possibly other columns

- are queried by downstream nodes to expedite processing

Marketing Automation Macro Variables

Here are some macro variables that are *automatically* available for use in a Marketing Automation stored process:

&INTABLE0	Stores a count of the number of input tables in the Process node.
&INTABLE1 – &INTABLEn	Hold the name of each upstream input MATables library data set that is linked directly to the Process node.
&INTABLE	Contains a space delimited list of all input tables.

29

Copyright © SAS Institute Inc. All rights reserved.



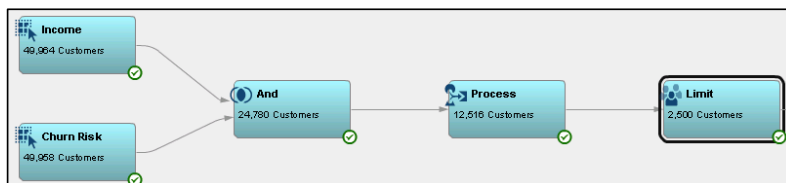
There are similarly named macro variables for output tables. For example, **&OUTTABLE1** through **&OUTTABLEn** hold the name of the table for each generated output cell.

Example 2: Apply a Propensity Model

```
data &outtable;
  set &intable;

  /* processing */
run;

%macnttab
```



30

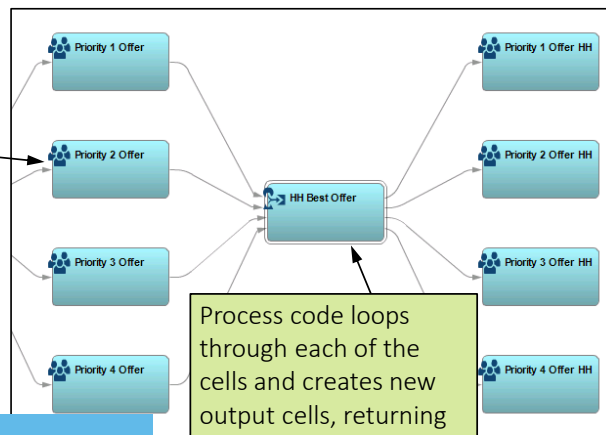
Copyright © SAS Institute Inc. All rights reserved.



If your stored process creates output tables in the **MATables** library, you must include a call to either **%Macnttab** or **%Maccount** to count the records in the output tables.

Example 3: Adjust Offers by Household

A campaign at the customer level assigns prioritized offers to customers. Customers in the same household could be assigned different offers.



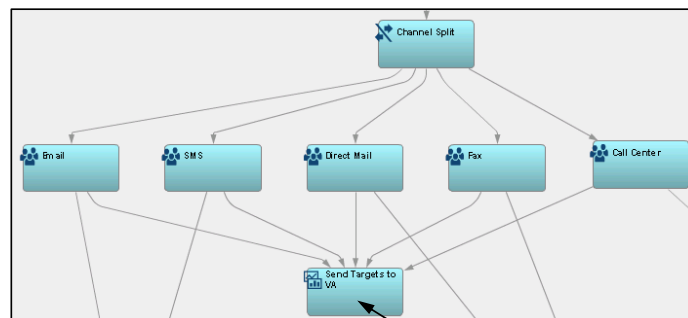
```
%do i=1 %to &intable0;
    /* Process each table */
%end;
```

Process code loops through each of the cells and creates new output cells, returning only the highest priority offer to qualifying customers in the household.



Copyright © SAS Institute Inc. All rights reserved.

Example 4: Create Reporting Data for Input Cells



Create a table for use in SAS Visual Analytics containing descriptive information about input cells.



Copyright © SAS Institute Inc. All rights reserved.

Example 4: Create Reporting Data for Input Cells

33

Copyright © SAS Institute Inc. All rights reserved.



Marketing Automation Stored Processes

- Your stored process can potentially reference temporary data sets specific to Marketing Automation.
- In order to create these data sets, your stored process code must include an appropriate call to the **%Maspinit** macro.

Data set	Contains information about
inputnodes	Nodes that immediately precede the Process node
outputnodes	Nodes that immediately follow the Process node
macrovar	Campaign, communications, exports, and output

- The following call creates all three data sets:
 - %maspinit(xmlstream= macrovar neighbor)

34

Copyright © SAS Institute Inc. All rights reserved.



You must also specify the corresponding data streams when registering the stored process.

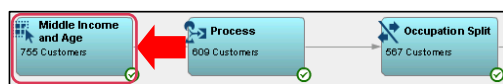
Columns in the inputnodes and outputnodes Data Sets

TABlename	The name of the data set in the MATables library for the node. This data set contains a list of subject IDs.
SUBJECTID	The subject ID for the node, such as a customer or household ID.
CODE	The code for the node used in Customer Intelligence Studio.
ID	Internal Customer Intelligence reference for the node.
DESCRIPTION	The Description field entry in Customer Intelligence Studio.
NAME	The Name field entry in Customer Intelligence Studio.

inputnodes Data Set

This data set contains information about the properties and **MATables** data sets for nodes that immediately precede the Process node.

TABlename	SUBJECTID	CODE	ID	DESCRIPTION	NAME
MATables.THCBDIS5BZHK54IREric	SUBJECT_ID_CUSTOMER		BEBEVZAJEFDDQHAD	Age 40-55 and Middle Income	Middle Income and Age



Middle Income and Age Properties

Name: Middle Income and Age

Description: Age 40-55 and Middle Income

Criteria:

Data Item	Value
Age	40 to 55
and	
Income Group	Middle

Rule: Any value meets these criteria

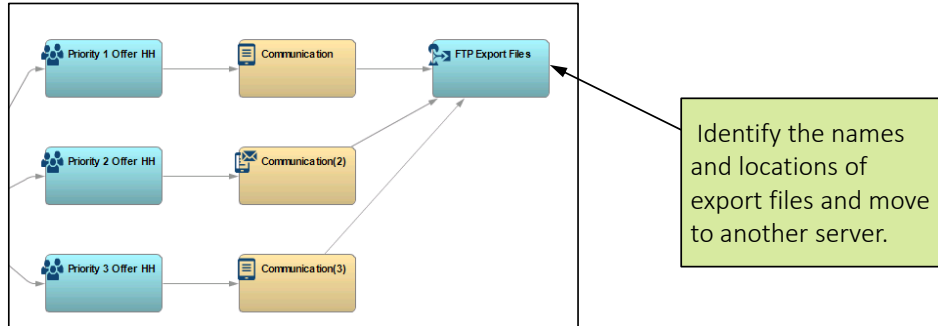
Count

Subject: Customers Total: 755 Last run on: Mar 11, 2016 10:44 AM

Update

Example 1: FTP Export Files

Requirement 3: Identify the location and name of export files.



37

Copyright © SAS Institute Inc. All rights reserved.



macrovar Data Set

- The **macrovar** data set contains information about the campaign and applicable communications, exports, and output.
- The table includes the following columns:

CATEGORY	Variable category
NAME	Variable name
DATATYPE	Variable data type (char, date, or numeric)
VALUE	Variable value
PARENT	Parent for the corresponding row details

38

Copyright © SAS Institute Inc. All rights reserved.



macrovar Table: Category

The value of the Category column reflects the type of information in that row of the table.

Value	Category
CAMPAIGNINFO	Campaign information
CAMPAIGNUDF	Campaign custom details
COMMUNICATIONINFO	Communication information
COMMUNICATIONUDF	Communication custom details
INPUTCELL	Communication input cell information
EXPORTINFO	Export information
CICOMMONINFO	Common data model information
OUTPUTINFO	Process node output tables and cells

39



Communication and export information is included in the data set only when a Communication node is upstream from the Process node.

macrovar Table: Export Information

Rows in the table with Category = EXPORTINFO provide information about exports for an upstream Communication node.



CATEGORY	NAME	DATATYPE	VALUE	PARENT
EXPORTINFO	EXPORT_NAME	Char	FIServ - Export to Delimited File	COMM687
EXPORTINFO	EXPORT_TYPE	Numeric	7	FIServ - Export to Delimited File
EXPORTINFO	EXPORT_PATH	Char	D:\CI\Common\Data\CIExport	FIServ - Export to Delimited File
EXPORTINFO	EXPORT_DESTINATION	Char	MyMAEExport.csv	FIServ - Export to Delimited File

40



Export information is included in the **macrovar** table only when a Communication node with one or more export definitions is upstream from the Process node. EXPORT_TYPE is a numeric constant indicating the export type, such as Excel or Delimited.

Stored Process Coding Best Practices

- ❶ Create SAS data sets in the temporary **Work** library.
 - This enables two or more campaigns to use the same stored process at the same time, without contention for a permanent library.
 - Each campaign has its own **Work** library, so there is no need to create separate data sets for each concurrent campaign.
 - Temporary data sets are automatically deleted.

41

Copyright © SAS Institute Inc. All rights reserved.



Stored Process Coding Best Practices

- ❷ Clear all library references as one of the last steps.

```
libname _ALL_ clear;
```



Failure to disconnect can cause stored process server issues.

42

Copyright © SAS Institute Inc. All rights reserved.



Stored Process Coding Best Practices

- ③ Disconnect from any database sessions that you open.

disconnect from oracle;



Failure to disconnect leaves connections open, which makes your database administrator unhappy.

43

Copyright © SAS Institute Inc. All rights reserved.



Stored Process Coding Best Practices

- ④ Include error handling code for efficient troubleshooting.

- Test for possible error conditions and write code to do the following:
 - end gracefully
 - produce relevant log messages
 - produce appropriate error dialog box messages in Customer Intelligence Studio

44

Copyright © SAS Institute Inc. All rights reserved.



continued...

Stored Process Coding Best Practices

- ⑤ Test stored process code before registration.
 - Begin by testing outside of Customer Intelligence Studio, in an environment such as SAS Enterprise Guide.
 - Some resources needed by your program, such as macro variables or data sets created during campaign execution, might not be available.
 - You can manually create these resources for testing purposes.

45

Copyright © SAS Institute Inc. All rights reserved.



Stored Process Coding Best Practices

- ⑤ Test stored process code before registration.
 - If testing there is successful, you might be able to test your code in Customer Intelligence Studio using a Process node with Type = Code.
 - In order to test in a Process node with Type = Code, the appropriate capabilities must be granted to you by an administrator.
 - Some stored process features such as creation of output cells are not supported for Type = Code.



Testing your code in a production environment is **not** recommended.

46

Copyright © SAS Institute Inc. All rights reserved.



Demonstration Scenario

- Create an output table by reading from an input table.
- Use the **macrovar**, **inputnodes**, and **outputnodes** data sets

```

/*This code is used to test stored process registration settings */
%maspinit(xmlstream=neighbor macrovar)

data &outtable1;
  set &intable1;
  /* Add code to manipulate output table */
run;

data _null_;
  if exist('work.macrovar') then putlog 'NOTE: The dataset WORK.MACROVAR exists';
  else putlog 'WARNING: The dataset WORK.MACROVAR does not exist';
  if exist('work.inputnodes') then putlog 'NOTE: The dataset WORK.INPUTNODES exists';
  else putlog 'WARNING: The dataset WORK.INPUTNODES does not exist';
  if exist('work.outputnodes') then putlog 'NOTE: The dataset WORK.OUTPUTNODES exists';
  else putlog 'WARNING: The dataset WORK.OUTPUTNODES does not exist';
run;

%MACount(&outtable1)
%Mastatus( &_stpwork.status.txt )

```

47

Copyright © SAS Institute Inc. All rights reserved.



Demonstration Scenario

- Required macro calls
 - %Stpbegin and %Stpend
 - %Maspinit
 - %Macount or %Macnttab
 - %Mastatus
- Required registration settings
 - MAUser keyword
 - Result capabilities: Package
 - macrovar and Neighbor data streams

48

Copyright © SAS Institute Inc. All rights reserved.





Registering a Stored Process Using SAS Enterprise Guide

1. Open a campaign.
 - a. Launch Customer Intelligence Studio and log on with the credentials **Eric** and **Student1**.
 - b. Open the selection campaign named **Ask the Expert Demo**.
 - c. Notice that the campaign includes a Select node and a Process node.
 - d. Open the Process node.
 - 1) Click **Select Process** to view the available stored processes. Notice there is not a stored process named **Ask the Expert Demo**.
 - 2) Click **Cancel**.
 - e. Click **Cancel**.
2. Launch Enterprise Guide and create a new project.
3. Open the program **D:\workshop\elmaacd\Registration_Demo_EG.sas**. Notice the following regarding the program:
 - a. It does not contain the required **%Stpbegin** and **%Stpend** calls.
 - b. It contains the required Marketing Automation macro calls **%Mastatus** and **%Macount**. (The latter is required because this stored process creates an output table in the **MATables** library referenced by **&OUTTABLE1**.)
 - c. It contains the call to **%Maspinit** required to create the **macrovar**, **inputnodes**, and **outputnodes** data sets.
 - d. It writes messages to the log regarding existence of these data sets.
 - e. It creates a single output table based on a single input table and does not do any actual processing of the corresponding subject list.

Note: This program is useful for demonstration purposes to confirm stored process registration settings, but it would not otherwise be useful in a campaign.
4. Right-click the program and select **Created Stored Process**. The Create New Stored Process Wizard appears.
5. Complete step 1 of the wizard (name and description).
 - a. Specify **Ask the Expert Demo** as the name.
 - b. If necessary, click the **Browse** button and navigate to **/CI/Financial Services/Stored Processes for Location**. This is the metadata folder used to store stored process information for a specific Customer Intelligence business context.
 - c. Enter **This stored process is for demonstration only** as the description.
 - d. Add the **MAUser** keyword.
 - 1) Click **Add keyword**.
 - 2) Type **MAUser**.

Note: The **MAUser** keyword is case sensitive.
 - e. Click **Next**.

6. Complete step 2 of the wizard (SAS Code).
 - a. The code included in the program appears. Notice again that the required calls to **%Stpbegin** and **%Stpend** are not included in the code.
 - b. Click **Include code for** at the bottom of the window. Confirm that **Stored process macros** is selected. This selection adds **%Stpbegin** to the beginning of the existing code and **%Stpend** to the end.
 - c. Click **Next**.
7. Complete step 3 of the wizard (Execution Options).
 - a. If necessary, select **SASApp** as the application server.
 - b. Click the down arrow next to Source code repository and select **D:\CI\Financial Services\Stored Processes**. This is the folder on the application server where stored process code is stored.
 - c. Confirm that **Ask the Expert Demo.sas** is specified as the source file.
 - d. Confirm that **package** is selected for **Result capabilities**.
 - e. Click **Next**.
8. Complete step 4 of the wizard (Prompts).

This stored process does not use prompts. Click **Next**.
9. Complete step 5 of the wizard (Data Sources and Targets).

Note: You might have to click **Next** and **Back** to access the Data Sources and Targets page.

 - a. Specify the macrovar stream.
 - 1) Click **New** for **Data Sources (input streams to a stored process)**. The Create a New Data Source window appears.
 - 2) Under Form of Data, select **XML based data**.
 - 3) Enter **macrovar** for **Fileref**.
 - 4) Enter **macrovar** for **Label**.
 - 5) Click **OK**.
 - b. Specify the Neighbor stream.
 - 1) Click **New** for **Data Sources (input streams to a stored process)**. The Create a New Data Source window appears.
 - 2) Under Form of Data, select **XML based data**.
 - 3) Enter **Neighbor** for **Fileref**.
 - 4) Enter **Neighbor** for **Label**.
10. Verify settings in step 6 of the wizard (Summary).
 - a. Confirm that keywords were entered properly.
 - b. Click **Show full SAS code** and notice that **%Stpbegin** and **%Stpend** were added to the code.
 - c. Clear the check box for **Run stored process when finished**.
 - d. Click **Finish**.

11. Verify stored process settings in Customer Intelligence Studio.
 - a. Return to Customer Intelligence Studio and the campaign named **Ask the Expert Demo**.
 - b. Configure the Process node.
 - 1) Double-click the **Process** node to open it.
 - 2) Select the stored process.
 - a) Confirm that **Process** is selected for **Type**.
 - b) Click the **Select Process** button.
 - c) Select the process named **Ask the Expert Demo**.
 - d) Click **OK**.
 - 3) Click **OK**.
 - c. Execute the process and view the log.
 - 1) Right-click the **Process** node and select **Execute**.
 - 2) Double-click the **Process** node to open it.
 - 3) Click the **Log** tab.
 - 4) Scroll down to find the notes confirming the existing of the **macrovar**, **inputnodes**, and **outputnodes** data sets.

NOTE: The dataset WORK.MACROVAR exists
NOTE: The dataset WORK.INPUTNODES exists
NOTE: The dataset WORK.OUTPUTNODES exists
 - 5) Click **Save** to close the Process node.
 - 6) Close the campaign and save the changes.

End of Demonstration

1.2 Learn More about Custom Processing

To Learn More about Custom Processing

Customer Intelligence

Engagement



<http://support.sas.com/training/us/paths/ci.html>