

Paper 339-2010

## How to Publish Your ODS Reports to a Predefined Channel

Emily (Yan) Gao, SAS Institute Inc., Beijing, China

### ABSTRACT

This presentation will show how to publish your ODS reports to a predefined channel, which is a repository that users can subscribe to. Any information published to a channel is available to all of that channel's subscribers. Each subscriber specifies how the new information is delivered. The advantage of publishing to a channel is that the publisher does not need to create or maintain a list of relevant recipients. Instead, users are responsible for subscribing to the correct channels for their work needs. Furthermore, you can share your ODS reports with non-subscribers, if the persistent storage of a channel is based on a WebDAV server. With any WebDAV server management tool, you can grant different access permissions to different groups to ensure information security. Some SAS® applications such as the SAS® Information Delivery Portal can connect to a WebDAV server and list its contents. Anyone who has permission to access the directory can view published ODS reports in SAS Information Delivery Portal.

### INTRODUCTION

The SAS® Publishing Framework of SAS® Integration Technologies provides a series of SAS CALL routines and application programming interfaces (APIs) to help users publish SAS files to a channel, but there is no routine or API to let users directly publish a directory to a channel. If users want to publish volumes of files stored under multi-layer directories to a channel, then it will not be an easy job.

For example, in Figure 1 the result folder is the root directory to store all ODS reports to be published. The first-layer subdirectory stands for project ID and the second-layer subdirectory stands for time. Directory names of the two layers are not fixed, and they are determined by the real data. In order to publish all SAS files under the result folder, users need to write SAS code to go through all subdirectories and publish each SAS file. And, most importantly, to ensure URL links in SAS files work correctly after the files are published, SAS files in the WebDAV server need to be organized in the same structure as the original directories.

This paper describes how to publish ODS reports to a WebDAV-based channel easily by running a SAS program, how to modify access permissions using a tool provided by SAS, and how to view the published ODS reports in the SAS Information Delivery Portal.

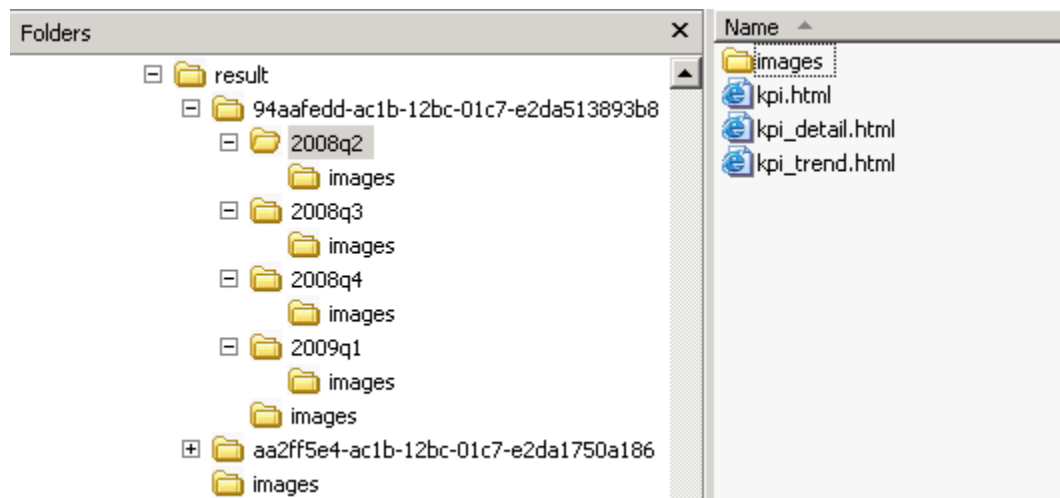


Figure 1. ODS reports to be published

## PUBLISH ODS REPORTS TO A WEBDAV-BASED CHANNEL

### CREATE ODS REPORTS

This is an example of code used to create ODS HTML outputs.

```
filename odsfile "C:\sgf2010\result\demo.html";  
ods html gpath="C:\sgf2010\result"(url='./') file=odsfile;
```

```

title "SGF 2010 Demo";
proc gchart data=sashelp.class;
    hbar age;
run;quit;

proc print data=sashelp.class;
run;quit;
ods html close;

```

## PUBLISHING FRAMEWORK OF SAS INTEGRATION TECHNOLOGIES

The Publishing Framework enables you to create packages that contain one or more information entities, including SAS data sets, SAS catalogs, SAS databases, and almost any other type of digital content. After creating a package, you can publish the package to a channel.

The following steps and their corresponding publish CALL routines are needed for core implementation:

### 1. Initialize a package.

```
CALL PACKAGE_BEGIN(packageId, desc, nameValue, rc<, properties, propValue1,
...propValueN>);
```

### 2. Insert a file into a package.

```
CALL INSERT_FILE(packageId, filename, filetype, mimeType, desc, nameValue, rc<,
properties, propValue1, ...propValueN>);
```

### 3. Insert HTML files into a package.

```
CALL INSERT_HTML(packageId, body, bodyUrl, frame, frameUrl, contents, contentsUrl,
page, pageUrl, desc, nameValue, rc<, properties, propValue1, ...propValueN>);
```

### 4. Insert a package into another package.

```
CALL INSERT_PACKAGE(packageId, insertPackageId, rc<, properties, propValue1,
...propValueN>);
```

### 5. Publish a package to a channel.

```
CALL PACKAGE_PUBLISH(packageId, publishType, rc, properties,<propValue1,
...propValueN>);
```

### 6. Free the resources that are associated with a package.

```
CALL PACKAGE_END(packageId, rc);
```

## PUBLISH ODS REPORTS

Now you only need two steps, which make publishing easier.

### 1. Submit the following code to compile SAS macros.

```

%macro chkrc(function);
    %if &rc = 0 %then %put "NOTE: &function succeeded.";
    %else
    %do;
        %let msg= %sysfunc(sysmsg());
        %put &function failed - &msg;
    %end;
%mend chkrc;

%macro mm_pack_directory(dir, dirname, spkid, delim, channel, subject, userid,
passwd);
    %let filrf = rptdir;
    %let rc = %sysfunc(filename(filrf));
    %let rc = %sysfunc(filename(filrf, &dir));
    %if &rc ne 0 %then %goto exit;

    %let dirid = %sysfunc(dopen(&filrf));
    %if &dirid = 0 %then %goto exit;

    %let memcount = %sysfunc(dnum(&dirid));

```

```

%if (&dirname ne ) %then
  %let desc = &dirname;
%else
  %let desc = %bquote(&dir);
%let namevalue = ;
%let &spkid = 0;
%let rc = 0;
%let package_options = ABSTRACT;
%let abstract_text= %bquote(Published by &userid);
%syscall package_begin(&spkid, desc, namevalue, rc, package_options,
abstract_text);
%chkrc(Package Begin);

%do i = 1 %to &memcount;
  %let subname = %sysfunc(dread(&dirid, &i));
  %let subdir = &dir.&delim.&subname;
  %let filrf = mytmp;
  %let rc = %sysfunc(filename(filrf));
  %let rc = %sysfunc(filename(filrf, &subdir));
  %if &rc ne 0 %then %goto exit;

  %let d = %sysfunc(dopen(&filrf));

  /* Add a file to the package file */
  %if &d = 0 %then
  %do;
    %let fileext = %sysfunc(lowercase(%sysfunc(scan(&subname, -1, '.'))));

    options minoperator;

    /* image file */
    %if (&fileext in png gif jpg) %then
    %do;
      %let filename = filename: &subdir;
      %let fileType = binary;
      %let mimeType = image/gif;
      %let desc = Image File;
      %let nameValue = ;
      %syscall insert_file(&spkid, filename, fileType, mimeType, desc,
nameValue, rc);
    %end;

    /* html file */
    %else %if (&fileext in htm html) %then
    %do;
      %let body = filename: &subdir;
      %let bodyURL = ;
      %let frame = ;
      %let frameURL = ;
      %let contents = ;
      %let contentsURL = ;
      %let page = ;
      %let pageURL = ;
      %let desc = &subname;
      %let nameValue = ;
      %syscall insert_html(&spkid, body, bodyURL, frame, frameURL,
contents, contentsURL, page, pageURL,
desc, nameValue, rc);
    %end;

    /* rtf file */
    %else %if (&fileext in rtf) %then
    %do;
      %let filename = filename: &subdir;
      %let fileType = binary;
      %let mimeType = application/msword;

```

```

        %let desc = RTF File;
        %let nameValue = %str(RTF=YES WORD=YES);
        %syscall insert_file(&spkid, filename, fileType, mimeType, desc,
nameValue, rc);
        %end;

        /* pdf file */
        %else %if (&fileext in pdf) %then
        %do;
            %let filename = filename: &subdir;
            %let fileType = binary;
            %let mimeType = application/pdf;
            %let desc = PDF File;
            %let nameValue = ;
            %syscall insert_file(&spkid, filename, fileType, mimeType, desc,
nameValue, rc);
        %end;

        /* other format file */
        %else
        %do;
            %let filename = filename: &subdir;
            %let fileType = text;
            %let mimeType = text/plain;
            %let desc = Text File;
            %let nameValue = ;
            %syscall insert_file(&spkid, filename, fileType, mimeType, desc,
nameValue, rc);
        %end;

        options nominoperator;

        %chkrc(Insert file);
    %end;

    /* Add a directory to the package file */
    %else
    %do;
        %let &spkid._i = &i;
        %let &spkid._c = &memcount;
        %let &spkid._did = &dirid;
        %let &spkid._dir = &dir;
        %let &spkid._subname = &subname;

        %let &spkid._&i = 0;
        %mm_pack_directory(&subdir, &subname, &spkid._&i, &delim, &channel,
&subject, &userid, &passwd);

        %let i = &&&spkid._i;
        %let memcount = &&&spkid._c;
        %let dirid = &&&spkid._did;
        %let dir = &&&spkid._dir;

        %let propname = NESTED_NAME;
        %syscall insert_package(&spkid, &spkid._&i, rc, propname,
&spkid._subname);
        %chkrc(Insert package);
    %end;

    %let rc = %sysfunc(filename(filrf));
    %let rc = %sysfunc(dclose(&d));
    %end;

    /* If current directory is the report root directory, then publish the package
to channel */
    %if (&spkid = rootid) %then
    %do;

```

```

%let pubtype = TO_SUBSCRIBERS;
%let properties = PARENT_URL, METAUSER, METAPASS, SUBJECT;
%let url = ;
%syscall package_publish(&spkid, pubType, rc, properties, url, userid,
passwd, subject, channel);
%chkrc(publish package);

%syscall package_end(&spkid, rc);
%chkrc(package end);
%end;

%exit:
%let rc = %sysfunc(dclose(&dirid));
%let rc = %sysfunc(filename(filrf));
%mend mm_pack_directory;

%macro mm_publish_to_channel(dir, channel, subject, metaserver, metaport, userid,
passwd, delim);
options metauser = "&userid";
options metapass = "&passwd";
options metaserver = &metaserver;
options metaport = &metaport;

%mm_pack_directory(&dir, , rootid, &delim, &channel, &subject, &userid,
&passwd);
%mend mm_publish_to_channel;

```

2. Set appropriate macro parameters and run the macro to publish the reports.

```

%mm_publish_to_channel
(
  C:\sgf2010\result\,
  DavChannel,
  SGF 2010 Demo,
  mmserver1.na.sas.com,
  8561,
  %str(mmserver1\mdlmgradin),
  %str({sas002}1D3211221C1104E218FEC4002945FBA308AB65D4),
  %str(\)
);

```

SAS macro %mm\_publish\_to\_channel is designed to help users publish SAS reports and its macro parameter descriptions as follows.

```

%mm_publish_to_channel
(
  dir,           /*Directory to be published.*/
  channel,      /*Predefined channel, where to save published reports.*/
  subject,      /*Email subject.*/
  metaserver,   /*Metadata server.*/
  metaport,     /*Metadata server port.*/
  userid,       /*User name used to access metadata server.*/
  passwd,       /*User password used to access metadata server.*/
  delim        /*Directory separator.*/
);

```

Subscribers will receive e-mail notifications after the reports are published.

**SGF 2010 Demo**

sbjgay@mmserver1.na.sas.com [sbjgay@mmserver1.na.sas.com]

**Sent:** Tuesday, August 04, 2009 6:50 PM

**To:** Emily (Yan) Gao

C:\sgf2010\result\

Published on August 04, 2009 21:50:11 -0400 to SAS Channel: DavChannel

Published by mmserver1\mdlmgradmin

<http://mmserver1.na.sas.com:8080/SASContentServer/repository/default/sasdav/sgf2010/se54080a/>

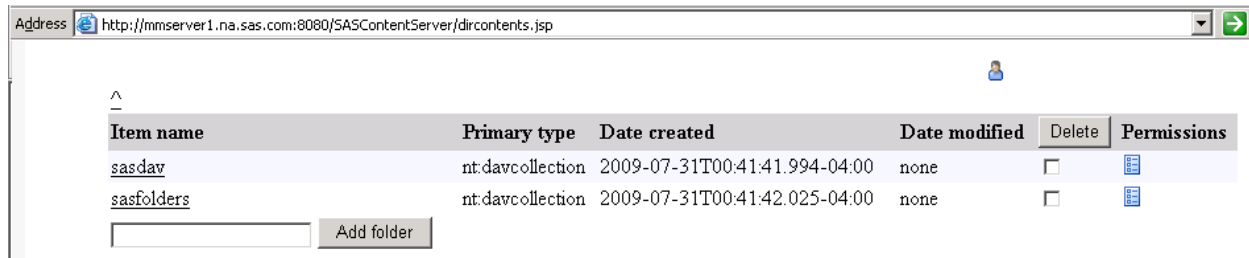
**Figure 2. E-mail notification**

**GRANT ACCESS PERMISSION**

For non-subscribers, you can grant them permission to access published reports following these steps:

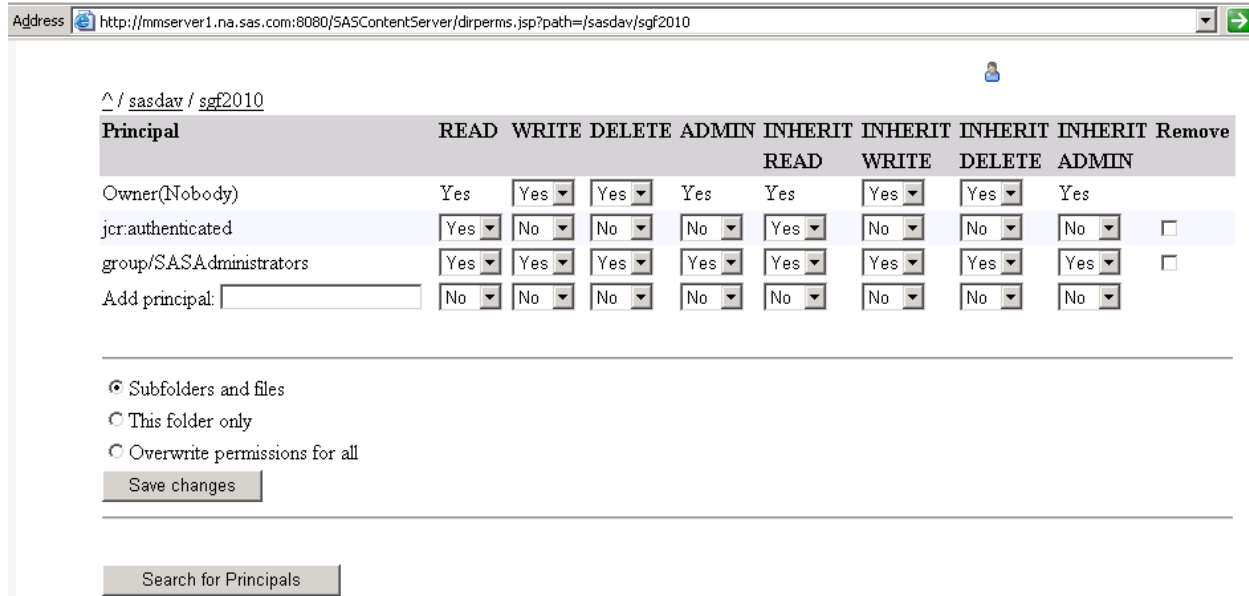
1. List all directories in WebDAV server.

http://yourserver:8080/SASContentServer/dircontents.jsp



**Figure 3. Directory list of WebDAV server**

2. Grant access permission.



**Figure 4. Permission management**

**VIEW ODS REPORTS IN THE SAS INFORMATION DELIVERY PORTAL**

A WebDAV Navigator portlet is a portlet that enables you to view files in a WebDAV repository. To view published reports, first you need to add a WebDAV Navigator portlet to one page.

### Add Portlets to Page • SGF\_Demo

Create Search

Portlet type: WebDAV Navigator

\*Name: my WebDAV Navigator

Description:

Keywords:

Add

Done

**Figure 5. Add a WebDAV Navigator portlet**

Once the portlet is added, you can see WebDAV folders.

### SAS® Portal

Home SGF\_Demo

my WebDAV Navigator

Location: WebDAV Up one level Show description

Name	Type	Date Modified
sgf2010	Folder	08/04/2009
Templates	Folder	07/31/2009
Users	Folder	08/04/2009

**Figure 6. WebDAV folders**

Drill down to the published package folder and you can see the ODS reports.

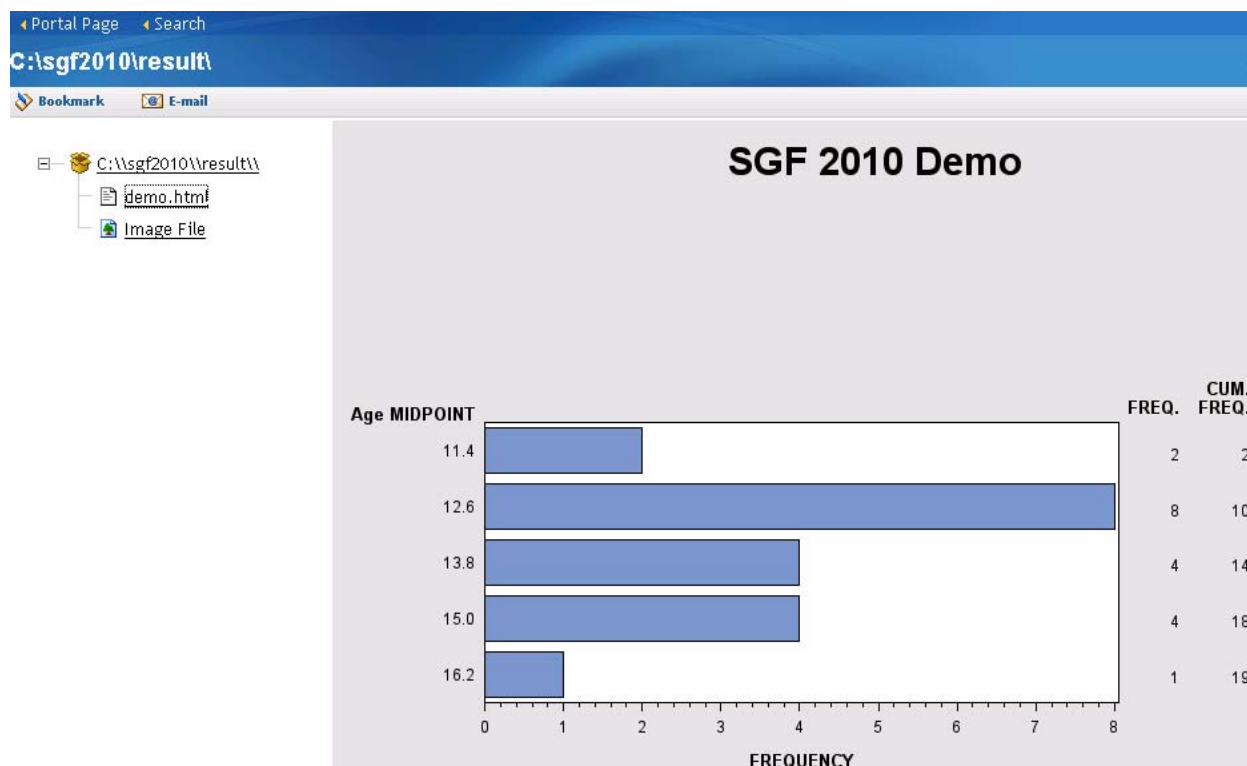


Figure 7. Published reports

## CONCLUSION

This paper describes how to publish your ODS reports to a channel and view the reports in the SAS Information Delivery Portal. Besides channels, users can also use the Publishing Framework of SAS Integration Technologies to publish ODS reports to an archive, e-mail, queues, or to a WebDAV server. To achieve that, users need to modify the parameter values of the routine syscall package\_publish in the macro %mm\_pack\_directory. Consequently, an ODS report can be distributed conveniently to various consumers.

## REFERENCES

- SAS Institute Inc. 2009. *SAS® 9.2 Publishing Framework: Developer's Guide*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2006. CopyToWebDAV.sas [Program]
- SAS Institute Inc. 2009. SAS Institute white paper. "SAS® Enterprise Guide® — A Roadmap,"

## ACKNOWLEDGMENTS

I would like to thank Alfred Liu, Robert Chu, and Jifa Wei for reviewing drafts of this paper and providing helpful suggestions.

## CONTACT INFORMATION

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

Emily (Yan) Gao, Manager of Analytics Development Group  
 SAS Institute Inc.  
 19/F North Wing, Central Office Tower, Junefield Plaza, No. 10 Xuanwumenwai Dajie Xuanwu District  
 Beijing, China 100052  
 Work Phone: +86 10 8319-3720  
 Fax: +86 10 6310-9130  
 E-mail: [yan.gao@sas.com](mailto:yan.gao@sas.com)

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.