



CHAPTER

43

Examples of Agent Services

- Example 1 - Define an Agent* 453
Example 2 - Retrieve an Agent Information 454
Example 3 - Run an Agent 455

Example 1 - Define an Agent

This example defines the agent REPORTS to run on Thursday evening at 8:30 p.m. It invokes the Preview window in which the user enters the SAS statements that are needed to run at the specified time.

```

init:

station_c =
  loadclass('sashelp.connect.station.class');
station_i = instance(station_c);

collection = 'acollection';
call send(station_i, '_open', collection, rc);

agent_c = loadclass('sashelp.connect.agent');
agent_i = instance(agent_c);

domain = 'tcp//mynode.xyz.com/domsvr';
call send(agent_i, '_setDomainInfo', domain,
          collection, rc, "", station_i);

  /* display Preview window so user can   */
  /* enter SAS statements to define       */
  /* to agent                             */
rc = preview('EDIT');

agentname = "reports";
description =
  'Agent that runs thursday night report.';

  /* location of where agent actually     */
  /* runs, which differs from domain name */
  /* which specifies the DOMAIN server    */
  /* where the agent is defined           */
runloc = 'tcp//mainnode.xyz.com/myserv';

```

```

        /* valid userid and password for the      */
        /* system on which the agent actually     */
        /* runs (run location)                    */
security = 'userid.password';

        /* notification queue name                */
nqueue = "checkq";

        /* set schedule list so that agent runs */
        /* every thursday at 8:30 p.m.         */
schedule = makelist();

rc = insertc(schedule, 20, -1, "RUN_HOUR");
rc = insertc(schedule, 30, -1, "RUN_MINUTE");
rc = insertc(schedule, 5, -1, "RUN_DOW");
call putlist(schedule, 'scheduled list' ,2);
call send(agent_i, '_defineAgent',
          agentname, rc, description,
          runloc, security, schedule,
          nqueue, "COMPLETE", "RETAIN");

        /* clear preview buffer                  */
rc = preview('clear');

return;

```

Example 2 - Retrieve an Agent Information

This example uses the notification queue to retrieve the agent name and runkey values. The runkey can then be used to retrieve the actual log and output from the agent run.

```

init:

station_c =
    loadclass('sashelp.connect.station.class');
station_i = instance(station_c);

collection = 'acollection';
domain = 'tcp//mynode.xyz.com/domsvr';
call send(station_i, '_open', collection, rc,
          domain);

        /* open the notification queue          */
queue_c = loadclass('sashelp.connect.queue');
queue_i = instance(queue_c);
call send(queue_i, '_open', station_i,
          "checkq", "FETCH", rc);

eventtype = '';
runkey=0;
hdrlist = makelist();

```

```

call send(queue_i, '_query', eventtype,
          msgtype, hdrlist, 0, rc);

if (rc eq 0) and (msgtype eq 65539) then do;
  agentname = getnitemc(hdrlist, 'AGENT_NAME');
  runkey = getnitemn(hdrlist, 'AGENT_RUN_KEY');
  description = getnitemc(hdrlist,
    'DESCRIPTOR');
  put agentname;
  put runkey;
  put description;
end;

rc = dellist(hdrlist);

/* close queue */
call send(queue_i, '_close', rc);

agent_c = loadclass('sashelp.connect.agent');
agent_i = instance(agent_c);
call send(agent_i, '_setDomainInfo', domain,
          collection, rc, "", station_i);

/* retrieve agent log and output spool */
/* using agent name and runkey retrieved */
/* from queue header information */
model_c =
  loadclass('sashelp.connect.runspool.class');
model_i = instance(model_c);
call send(agent_i, '_retrieveAgentRunInfo',
          agentname, runkey, rundt, runcc, rc,
          model_i, model_i);

/* now model_i can be used to display log */
/* and output from agent run by attaching */
/* to text viewer object */
return;

```

Example 3 - Run an Agent

This example runs an agent that is already defined at the DOMAIN server. It then uses the runkey that is returned by the `_RUN_AGENT_` method to retrieve the log and output from the agent run.

```

init:

station_c =
  loadclass('sashelp.connect.station.class');
station_i = instance(station_c);

collection = 'acollection';
call send(station_i, '_open', collection, rc);

```

```

agent_c = loadclass('sashelp.connect.agent');
agent_i = instance(agent_c);

domain = 'tcp//mynode.xyz.com/domsvr';
call send(agent_i, '_setDomainInfo', domain,
          collection, rc, "", station_i);

    /* name of agent already defined at the      */
    /* domain server                            */
agentname = "reports";
description =
    'Unscheduled run of thursday night agent';

    /* location of where agent actually          */
    /* runs, which differs from domain name     */
    /* which specifies the DOMAIN server where  */
    /* the agent is defined                     */
runloc = 'tcp//mainnode.xyz.com/myserv';

    /* valid userid and password for the system */
    /* on which the agent actually runs (run    */
    /* location)                                */
security = 'userid.password';

runkey=0;
call send(agent_i, '_runAgent', agentname,
          runkey, rc, description, runloc, security);

if (rc eq 0 ) then do;
    model_c =
        loadclass('sashelp.connect.runspool.class');
    model_i = instance(model_c);

        /* retrieve agent log and output spool */
        /* using agent name and runkey that are */
        /* returned by run_agent method        */
    call send(agent_i, '_retrieveAgentRunInfo',
              agentname, runkey, rundt, runcc,
              rc, model_i, model_i);

        /* now model_i can be used to display log */
        /* and output from agent run by attaching */
        /* to text viewer object                  */
end;

return;

```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/CONNECT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 537.

SAS/CONNECT User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-477-2

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.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

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.

IBM[®], AIX[®], DB2[®], OS/2[®], OS/390[®], RS/6000[®], System/370[™], and System/390[®] are registered trademarks or trademarks of International Business Machines Corporation. ORACLE[®] is a registered trademark or trademark of Oracle Corporation. [®] indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.