



CHAPTER

42

Call Routine Interface to Agent Services

Introduction 449

Dictionary 449

Introduction

A CALL routine interface has been implemented to provide access to the agent scheduling services. This interface is included with SAS/CONNECT software and can be used within a SAS DATA step or a SAS Macro Facility.

This interface provides the user with CALL routines that can run a particular agent.

Dictionary

AGENT_RUN

Run an agent defined to the DOMAIN server.

Syntax

```
CALL AGENT_RUN(stationId, '_runAgent', agentName, runKey, rc<, description,  
runLocation, securityInfo, qname>);
```

Where...	Is type...	And represents...
<i>stationId</i>	N	station identifier
<i>agentName</i>	C	name of agent to run
<i>runKey</i>	N	run instance key
<i>rc</i>	N	return code
<i>description</i>	C	optional text description
<i>runLocation</i>	C	optional host location to run agent

Where...	Is type...	And represents...
<i>securityInfo</i>	C	optional security information
<i>qname</i>	C	optional queue name

This function runs an agent that is already defined to the DOMAIN server. There is no way to define an agent through the functional interface; this must be done in an SCL application.

The *stationId* is the station identifier of a successfully opened station. The function *station_open* must have been invoked successfully prior to this call.

The *agentName* parameter is a name of an agent that has already been defined by using the SCL method *_defineAgent*.

The *runKey* parameter is set when the agent is run successfully. In an SCL environment, *runKey* can then be used as an input parameter to *_abortAgentRun*, *_retrieveAgentRunInfo*, or *_purgeAgentRunInfo*. This value is of no subsequent use within the functional environment because there are no other functions supported at this time.

The *rc* return code identifies whether the agent was run successfully.

The *description* parameter is an optional user-specified text description of the agent. This string is supplied with run notifications as a presentation context aid.

The *runLocation* parameter identifies the host and spawner on which the agent should actually run when it executes. The agent may be defined to the DOMAIN server that exists on host A, but the agent may actually be defined to execute on host B.

The *securityInfo* parameter is the *userid.password* specification that is needed when the agent actually executes on a given host. This is a required parameter if the target *runlocation* is a secured environment, otherwise it's optional.

Note: This may be different than the *userid.password* specified on the *_setDomainInfo* SCL method. If different, *userid.password* is used to communicate with the DOMAIN server when running secured. If the same, it is used to launch the agent on the specified host by using *runLocation*. △

The *qname* parameter, if specified, is the name of the domain message queue to which the agent run completion notification is delivered.

Note: All optional parameters must be specified in the order stated by the syntax. △

Example

This program opens the station interface TESTDMM and runs the agent REPORT. Notice that the results are sent to the message queue RQUEUE.

```
data _null_;

    /* open the station */
```

```

stid=0;
stname="TESTDMM";
rc=0;
domain="/shr1";
security="user1.pass1";
call station_open(stid, stname, rc, domain, security);

if rc = 0 then put 'Station open successful.';
else do;
    msg = sysmsg();
    put msg;
end;

    /* Run REPORT agent that is already defined */
    /* to the DOMAIN server. Notice that the */
    /* only optional parameter needed is QNAME, */
    /* but the others are specified as NULL */
    /* because the optional parameters are */
    /* positional and must be specified. */
agName="report";
runkey=0;
rc=0;
desc="";
runloc="";
sec="";
qn="rqueue";
call agent_run(stid, agName, runkey, rc, desc,
               runloc, sec, qn);

if rc = 0 then do;
    put 'Agent run successful.';
    put 'Runkey is ' runkey best32.;
end;
else do;
    msg = sysmsg();
    put msg;
end;

run;

```


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.