

SAS[®] 9.3 Qualification Tools User's Guide



Copyright Notice

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., SAS® 9.3 *Qualification Tools User's Guide*, Cary, NC: SAS Institute Inc., 2011.

SAS® 9.3 Qualification Tools User's Guide

Copyright © 2011, 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, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc. Limited permission is granted to store the copyrighted material in your system and display it on terminals, print only the number of copies required for use by those persons responsible for installing and supporting the SAS programming and licensed programs for which this material has been provided, and to modify the material to meet specific installation requirements. The SAS Institute copyright notice must appear on all printed versions of this material or extracts thereof and on the display medium when the material is displayed. Permission is not granted to reproduce or distribute the material except as stated above.

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.

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.

Table of Contents

Introduction	1
SAS Installation Qualification Tool (SAS IQ)	3
Overview	3
Windows	3
Running the Installation Qualification Tool on Windows Systems	3
Output	4
UNIX	4
Running the SAS Installation Qualification Tool on UNIX Systems	4
Reviewing the Results of the SAS Installation Qualification Tool	4
z/OS	4
Running the SAS Installation Qualification Tool on z/OS Systems	4
Input	5
Process	5
Output	5
Launching	5
SAS Operational Qualification Tool (SAS OQ)	7
Overview	7
Windows	7
Directory Structure	7
Running the Operational Qualification Tool on Windows Systems	7
UNIX	7
Directory Structure	7
Running the SAS Operational Qualification Tool on UNIX Systems	8
SAS OQ Output for Both Windows and UNIX Systems	8
Output	8
Outdir Directory Structure	8
Reviewing the Results of the SAS Operational Qualification Tool	9
SAS Operational Qualification Tool Table Language	9
Table Language General Rules	9
Block Identifier	10
&test	10
Command	10
&rc	10
&run	11
&set	11
&unset	12
Macro	13
&infile	13
&outfile	13
SAS Operational Qualification Tool Command Line	14

Introduction

There are two qualification tools available with SAS 9.3—the SAS Installation Qualification Tool (SAS IQ) and the SAS Operational Qualification Tool (SAS OQ). These tools will assist you in qualifying the use of the SAS software in regulated industries. Together, these tools support the qualification aspect of the essential migration, integration, and verification processes customers need to move from previous versions of SAS to later releases.

SAS IQ assists you in demonstrating the SAS System has been installed and maintained to the manufacturer's specifications. SAS IQ verifies the integrity of each file in the SAS System 9 and provides the customer a set of reports detailing the results. SAS IQ is supported on all Windows, UNIX, and z/OS platforms for SAS 9.3. See “SAS Installation Qualification Tool (SAS IQ)” on page 3 for a detailed description.

SAS OQ assists you in demonstrating the SAS System is operational. SAS OQ uses SAS programs provided by the component development groups and will execute, process, and report the program results. SAS OQ is supported on all Windows and UNIX platforms for SAS 9.3. See “SAS Operational Qualification Tool (SAS OQ)” on page 7 for a detailed description.

***Note:** SAS OQ is not supported on z/OS. If you are installing SAS software on a z/OS host, after all installation jobs have finished, submit the VALID job in the CNTL data set and ensure that all steps complete with a return code of 0. Please refer to the installation instructions you received with your SAS Software Order E-mail for more information.*

SAS Installation Qualification Tool (SAS IQ)

Overview

The SAS Installation Qualification Tool (SAS IQ) validates a SAS System installation by verifying that each installed file is correct. A report is generated detailing all file results. This determination is made using the md5 algorithm to create a value for each file.

Important: SAS IQ validation represents the state of the system as it was at the time the installation completed. Most manual modifications made by the user to the SAS software components after the initial installation will be detected by SAS IQ validation. Installing software after an initial installation (such as an add-on installation or applying maintenance) should **not** result in additional files being flagged by SAS IQ as being changed.

For example, user modifications to configuration files such as sasv9.cfg or INI files will be flagged by SAS IQ as files that are now different in checksum value than the files originally installed on your system. In addition, renewing your SAS license will cause SAS IQ to flag files that have changed such as the core.sas7bcat file.

Windows

Running the Installation Qualification Tool on Windows Systems

There are two ways to run SAS IQ on a successful SAS 9.3 installation:

- From the start menu as follows:

```
Start -> Programs -> SAS -> Utilities -> SAS Install Qualification Tool 9.3
```

or

- Open a command line in DOS and type the following command:

```
<SASHOME>\InstallMisc\utilities\installqual\9.3\sasiqt.exe -out <dir>
```

In this command, <SASHOME> is the location where SAS Software is installed, and <dir> is the writable directory where the report file should be written to.

The -out option requires a writable directory passed with it in order for the tool to write the file to the specified location. You can specify the -out option on the command line when run via DOS prompt, or you can update the sasiqt.ini file to include the -out option argument by adding an additional argument to the .ini file. For example:

```
AddtlArgs_2=-out "C:\temp"
```

To run in headless mode, add the -silent option to the command line or as an additional argument in the sasiqt.ini file found in the same location as the sasiqt.exe executable. For example:

```
AddtlArgs_3=-silent
```

Output

SAS IQ will write the checksum validation results to an html file named `ValidationReport_<date-time>.html`. The html file simply formats the results and uses color-coded dividers in the form of boxes in the report to show failed and passed file checks.

By default, the html file is written to `<SASHOME>/InstallMisc/InstallValidation`.

The "-out" option allows users to redirect the html report file to an alternate location. The option must be followed by a valid writable path and can be passed via the command line or as an additional argument (`AddtlArg_#`) inside the `sasiqt.ini` file.

UNIX**Running the SAS Installation Qualification Tool on UNIX Systems**

To run SAS IQ on a successful SAS 9.3 installation, do the following:

From a console or a terminal window opened on the UNIX system, type the following command:

```
<SASHOME>\InstallMisc\utilities\installqual\9.3\sasiqt -out <dir>
```

Or, to run in headless silent mode:

```
<SASHOME>\InstallMisc\utilities\installqual\9.3\sasiqt_console -silent -out <dir>
```

Reviewing the Results of the SAS Installation Qualification Tool

SAS IQ will write the checksum validation results to an html file named `ValidationReport_<date-time>.html`. The html file simply formats the results and uses color-coded dividers in the form of boxes in the report to show failed and passed file checks.

By default, the html file is written to `<SASHOME>/InstallMisc/InstallValidation`.

The "-out" option allows users to redirect the html report file to an alternate location. The option must be followed by a valid writable path and can be passed via the command line or as an additional argument (`AddtlArg_#`) inside the `sasiqt.ini` file.

SAS IQ validates a SAS System installation by verifying that each installed file is correct. A report is generated detailing all file results. The status determination is made using the md5 algorithm to create a value for each file.

z/OS**Running the SAS Installation Qualification Tool on z/OS Systems**

To run SAS IQ on a successful SAS 9.3 installation, do the following:

The rexx script used to launch the SAS IQ Tool must be run in a z/OS shell environment. Using UNIX System Services on z/OS, this command can be issued as:

```
<SASHOME>/InstallMisc/utilities/installqual/9.3/sasiqt.rexx -out <dir>
```

Input

SAS IQ will read the `<SASHOME>/InstallMisc/InstallLogs/ValidationFileList.xml` data using Java API for XML Binding (JAXB) to get a list of files installed during deployment and their checksum values. The `ValidationFileList.xml` data is stored as:

```
<sasfile prodcode="base"
      checksum="2a998d9f46d441e30aee33b72fa89733"
      name="\path\relative\to\<SASHOME>" />
```

The `ValidationFileList.xml` file is written to the default log location, `<SASHOME>/InstallMisc/InstallValidation`, by the Install Tool at post install.

Process

For each file read from the Master File List, SAS IQ finds the installed file in `<SASHOME>`, gets the checksum data, and compares the checksum to test if it is equal to the original checksum from `ValidationFileList.xml`. Each checksum test produces a status. The status can be PASS, CHECKSUMWRONG, CHECKSUMMISSING, FILEMISSING, or UNKNOWN. Status data is recorded for each checked file.

- If the pre-install checksum data from `ValidationFileList.xml` and post-install checksum data matches for a file, the status is set to PASS.
- If the pre-install checksum data from `ValidationFileList.xml` and post-install checksum data does not match, the status is set to CHECKSUMWRONG.
- If the installed file does not have a checksum value, the status is set CHECKSUMMISSING.
- If SAS IQ is unable to find the file, the status is set to FILEMISSING.
- Other checksum failures are captured as UNKNOWN until we are able to classify them.

Output

SAS IQ will write the checksum validation results to an html file named `ValidationReport_<date-time>.html`. The html file simply formats the results and separates the results in color-coded divisions in the form of outlined boxes to show failed and passed file checks

By default the html report file is written to `<SASHOME>/InstallMisc/InstallLogs`. The "-out" option allows users to redirect the html report file to an alternate location. The option must be followed by a valid writable path and can be passed via the command line or as an additional argument (`AddtlArg_#`) inside the `sasiqt.ini` file.

Launching

SAS IQ is installed during deployment into the `<SASHOME>/InstallMisc/utilities/installqual/{SAS Version}` directory. Executable launchers named `sasiqt.exe`, `sasiqt`, and `sasiqt.rexx` are provided on the Windows, UNIX, and MVS platforms respectively.

SAS IQ displays a GUI progress monitor/bar as it validates files. To run in headless mode, enter the `-silent` option when launching on the command line or add `-silent` as an Additional Argument (AddtlArgs_) property in the `.ini` file. Running in headless mode on UNIX also requires adding the `mode=console` property setting under the `[default]` heading in `sasiqt.ini` in addition to using the `-silent` option.

SAS Operational Qualification Tool (SAS OQ)

Overview

Note: SAS OQ is not supported on z/OS. If you are installing SAS software on a z/OS host, after all installation jobs have finished, submit the VALID job in the CNTL data set and ensure that all steps complete with a return code of 0. Please refer to the installation instructions you received with your SAS Software Order E-mail for more information.

SAS Operational Qualification Tool (SAS OQ) is the program that invokes the operational tests. These tools and testware are shipped on SAS 9.3 media.

Use the platform-specific instructions which follow to run SAS OQ tests.

Windows

Directory Structure

SAS Operational Qualification Tool - `<SASROOT>\sastest\sasq.exe`

SAS Operational Qualification Tool support files - `<SASROOT>\sastest\`

Component content - `<SASROOT>\sastest<component>`

Running the Operational Qualification Tool on Windows Systems

To run SAS OQ on a successful SAS 9.3 installation, do the following:

- Open a command line in DOS and type the following command:

```
c:\><SASROOT>\sastest\sasq.exe -tables *:<component> -outdir <dir>
```

*Note: *.* runs all of the tables so you don't have to run each component individually.*

In this command, `<SASROOT>` is the location where the SAS System was installed, `<component>` is the installed component in the `sastest` directory for which you want to run tests, and `<dir>` is a directory to store the results.

UNIX

Directory Structure

SAS Operational Qualification Tool - `<SASROOT>/sastest/sasq.sh`

SAS Operational Qualification Tool support files - `<SASROOT>/sastest/`

Component content - `<SASROOT>/sastest/<component>`

Running the SAS Operational Qualification Tool on UNIX Systems

To run SAS OQ on a successful SAS 9.3 installation, do the following:

From a console or a terminal window opened on the UNIX system, type the following command:

```
<SASROOT>/sastest/sasoq.sh -tables *:<component> -outdir <dir>
```

*Note: *.* runs all of the tables so you don't have to run each component individually.*

In this command, <SASROOT> is the location where the SAS System was installed, <component> is the installed component in the `sastest` directory for which you want to run tests, and <dir> is a directory to store the results.

SAS OQ Output for Both Windows and UNIX Systems

SAS OQ generates XML data files to hold the raw results of the validation process. The SAS System is used to translate the XML data into PDF and HTML reports. The PDF and HTML reports are identical. The PDF can be used to create hard copies of the resulting validation. The HTML is easier to browse.

In all cases, the output report is organized first at the SAS System component level, and then by each component's files.

Output

SAS OQ generates XML data files to hold the raw results of the validation process. The SAS System is used to translate the XML data into PDF and HTML reports. The PDF and HTML reports are identical. The PDF can be used to create hardcopies of the resulting validation. The HTML is easier to browse.

In all cases, the output report is organized first at the SAS System component level, and then by each component's files.

Outdir Directory Structure

After you run SAS OQ, a result directory is created in the output directory you specified with `-outdir <dir>`, with the format `ftt_<YYYYMMDD>.000`. The trailing `.000` would be incremented for tool invocations on the same day. The following table describes the structure of the directory created with the `-outdir` option:

File Name	Description
<Component>	When tests are run, a directory is created for each to store files created by the tests.
Data	Directory containing the raw validation data for each component in XML format
html	Directory containing HTML files for each component
autoexec.sas	SAS file used in producing PDF and HTML output from XML data
index.xml	XML index file for all component summary
index.xsl	XML style sheet
sasoq.htm	HTML file containing index to summary and component information
sasoq.log	SAS log file created from running the autoexec.sas program

File Name	Description
sasoq.pdf	PDF file containing all summary and component information
SASOQindex.map	Intermediate file used to output XML to PDF and HTML
SASOQtest.map	Intermediate file used to output XML to PDF and HTML
sasoqtoc.htm	HTML file used to display component information
Test.xsl	XML style sheet

Reviewing the Results of the SAS Operational Qualification Tool

You can review SAS OQ results by opening the `sasoq.htm` file in a Web browser or by opening the `sasoq.pdf` file in Adobe Reader.

SAS Operational Qualification Tool Table Language

The primary input files for SAS OQ are test tables (stored in files with a `.tab` extension). These files define which tests to run, how to run the tests, and how to determine if the tests worked correctly.

Table Language General Rules

The following are basic rules for test table files:

- SAS OQ generally ignores white space (spaces or line breaks).
- Comments are delimited by `/*` and `*/`. They can span multiple lines and appear anywhere in a line. These comments work the same as C comments.
- Statement arguments fall into one of the following groups:
 - A **list** is a group of items in any order separated by commas. An example of a list is the set of options passed to the program being tested. The order of such options is irrelevant to the operation of SAS OQ. (The program being tested might require them to be in a particular order, but SAS OQ does not.)
 - A **structure** is a group of items in a specific order separated by colons. An example of a structure is a file specification. The elements must be in a particular order, so that SAS OQ can tell which part is the file name, which part is the component name, etc.
- Statements are processed in the order in which they occur in the test block, so the statement to run the test must be before the statements to process the results.
- Tables are made up of comments, `&set` commands, `&unset` commands, and test blocks. Each test block starts with the block identifier, `&test`, and contains a set of commands—a single `&run` followed by `&rc`. Within the commands you can optionally use the macros `&infile` and `&outfile`.

Block Identifier

The only valid block identifier is `&test`.

`&test`

Usage:

```
&test testname { test_stmt_list }
```

Arguments:

- o *testname* is the name of this test.
- o *test_stmt_list* is the list of statements that make up the test.

Description:

The `&test` statement defines a test block. There must be one test block for each test in a test table file.

Example:

See the complete example below.

Command

The commands are `&rc`, `&run`, `&set` and `&unset`.

`&rc`

Usage:

```
&rc ( value )
```

Arguments:

value is the expected value of the return code for this test.

Description:

The `&rc` statement tests the return code from the command specified in the `&run` statement. If the return code does not match the specified value, the test is marked as a failing test.

The use of the `&rc` command is required. You may not use it more than once per test block.

Example:

```
&rc( 0 )
```

&run

Usage:

```
&run ( verb : ( arguments ) )
```

Arguments:

- *verb* is the name of the program to run. In the initial version, the only verb allowed is "sas".
- *arguments* is the list of options and arguments to the program being run.

Description:

The `&run` statement runs the actual test command. There must be **one** `&run` command in each test block. The `&run` must be followed by `&rc`.

Example:

```
&run( sas : (-sysin &infile() -ls 78) )
```

This command runs the SAS system. The `&infile` statement (see below) passes an input file with the same base name as the test name. The other options are passed through unmodified.

&set

Usage:

```
&set ( name : value )
```

Arguments:

- *name* is the name of the variable you are defining consisting only of alphanumeric characters.
- *value* is the any sequence of tokens. If the sequence contains parentheses, they must be balanced or the unbalanced parenthesis needs to be escaped (see example).

Description:

This command allows you define variables and assign values to them. The names and values you define using `&set` are global and remain in effect for the remainder of the current.

You may change the value of a variable by assigning a new value later using another `&set` statement.

Once you define a variable, you can use it anywhere in the table (as long as the `&set` statement precedes its use). To use the variable you created, use `@name` to insert the value of the variable. If you use a variable that has not been defined, an empty string is used for the value.

All parentheses inside the variable value must be balanced. If you require unbalanced parentheses, use the `'\'` escape character to escape the parenthesis.

The `&set` statement may not be used inside of a block.

The `testname` and `tablename` variables are predefined for you and are set to the name of the test (or table) at the time the variable is expanded.

If the `@` sign is needed in the value of the variable but should not be treated as a variable expansion, then escape the `@` sign. See the examples below for a sample use of this.

Example:

```
&set( sasopts : -ls 78 -ps 60 )
```

This command sets the `sasopts` variable to “-ls 78 -ps 60”.

```
&set( infile : -sysin &infile( sas : @testname : base) )
```

This command sets the `infile` variable to “-sysin &infile(sas : @testname : base)”.

```
&set( infile1 : -sysin &infile\( sas : )
&set( infile2 : : base\) )
&set( run : sas : ( @infile1 @testname @infile2 \@noexpand) )
```

This command sets the `infile1` variable to `-sysin &infile(sas :, infile2 to : base)` and `run` to `sas : (@infile1 @testname @infile2 @noexpand)`. Assuming a test name of `test1`, the `run` variable will expand to `sas : (-sysin &infile(sas : test1 : base)`.

&unset

Usage:

```
&unset ( name )
```

Arguments:

- o *name* is the name of a variable that may, or may not, have been set using the `&set` command. Note that the `@` is **not** included when specifying the variable name, as that would cause the variable to be replaced with its value before the `unset` processing occurs.

Description:

This command allows you to guarantee that a variable is no longer set and if used, will expand to nothing.

Example:

```
&unset( testopts )
```

This command clears the `testopts` variable even if it had not previously been defined.

Macro

The macros are `&infile` and `&outfile`.

&infile

Usage:

```
&infile( ext [: name [: component]]) or
&infile([ext]: name [: component]) or
&infile([ ext ] : [name] : component)
```

Arguments:

- o *ext* is the file extension. If it is omitted, the default is "sas".
- o *name* is the file name. If it is omitted, the name of the test is used as the file name.
- o *component* is the component area where the file is installed. If it is omitted, the component area where the table file was found is used.

Description:

The `&infile` statement defines an input file. It is normally used as part of an option to a command in the `&run` statement. Note that it is allowed to omit all of the arguments to `&infile`.

Example:

```
&infile( sas : foobar : base )  

  Specifies the file named foobar.sas from the base testware area.
```

```
&infile()  

  Specifies the file testname.sas (where testname is the name of the current test) from the testware area of the component where the .tab file is located.
```

&outfile

Usage:

```
&outfile( ext [: name ])
```

Arguments:

- o *ext* is the file extension. There is no default for the extension. It must be specified.
- o *name* is the file name. If it is omitted, the name of the test is used as the file name.

Description:

The `&outfile` statement defines an output file. It is normally used as part of an option to a command in the `&run` statement.

```
&outfile( lst : foo )
```

Specifies the file named `foo.lst`.

```
&outfile(log)
```

Specifies the file `testname.log`.

SAS Operational Qualification Tool Command Line

SAS OQ command line arguments are defined as follows:

```
-tables table_spec [table_spec ...]
```

The `-tables` option specifies which test tables to run. A `table_spec` has the form `name:component`. This is very similar to the specifications in the `&infile` statement, except that the extension is always assumed to be `.tab`. At least one `table_spec` is required.

```
-help
```

The `-help` option requests the tool to print its usage.

```
-verbose | -quiet
```

These two arguments dictate the amount of STDOUT output SAS OQ writes during execution. The `-verbose` option tells SAS OQ to print extra output to STDOUT when executing. This is useful when setting up new tests. The `-quiet` option tells SAS OQ to only report errors. This is useful when scripting SAS OQ commands.

```
-sasroot <path>
```

This argument tells SAS OQ where to locate The SAS System installation. By default, the tool will attempt to locate the SASROOT path itself. If it is unable to, for some reason, this argument can be used to explicitly set that path. This can also be used to test different SAS System installations on the same machine.

```
-testware <path>
```

This argument specifies the location of the testware. By default `SASROOT/sastest` is used.

```
-outdir <path>
```

This argument specifies the location where all output files should be written. The default is to use the current directory where the tool execution took place.



THE
POWER
TO KNOW.

support.sas.com

SAS is the world leader in providing software and services that enable customers to transform data from all areas of their business into intelligence. SAS solutions help organizations make better, more informed decisions and maximize customer, supplier, and organizational relationships. For more than 30 years, SAS has been giving customers around the world The Power to Know®. Visit us at **www.sas.com**.